

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE  
UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À  
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE  
À L'OBTENTION DE LA  
MAÎTRISE EN GÉNIE DE LA PRODUCTION AUTOMATISÉE  
M. Ing.

PAR  
Joe Sfeir

NAVIGATION D'UN ROBOT MOBILE EN ENVIRONNEMENT INCONNU UTILISANT  
LES CHAMPS DE POTENTIELS ARTIFICIELS

MONTRÉAL, LE 26 NOVEMBRE 2009

© <Joe Sfeir, 2009>

**PRÉSENTATION DU JURY**  
**CE MÉMOIRE A ÉTÉ ÉVALUÉ**  
**PAR UN JURY COMPOSÉ DE**

M. Maarouf Saad, directeur de mémoire  
Département de génie électrique à l'École de technologie supérieure

M. Hamadou Saliah-Hassane, codirecteur de mémoire  
Téléuniversité/UQAM

M. Vahé Nerguizian, président du jury  
Département de génie électrique à l'École de technologie supérieure

M. Cédric Mélançon, examinateur externe  
CMC ÉLECTRONIQUE INC.

**IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC**

**LE 26 NOVEMBRE 2009**

**À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE**

## **REMERCIEMENTS**

Je tiens à remercier M. Maarouf Saad qui m'a dirigé dans ce travail avec beaucoup de patience et de compréhension. Je remercie aussi M. Hamadou Saliah-Hassane pour son soutien solide et amical.

Je tiens surtout à remercier Charles Fallaha, Claude Kaddissi, Élie Maalouf et Fawzia El-Kassir pour leur soutien inconditionnel et pour leur esprit critique qui m'a aidé à approfondir et perfectionner ce projet.

# NAVIGATION D'UN ROBOT MOBILE EN ENVIRONNEMENT INCONNU UTILISANT LES CHAMPS DE POTENTIELS ARTIFICIELS

Joe Sfeir

## RÉSUMÉ

Ce projet traite de la navigation d'un robot mobile en environnement inconnu. Le but est de développer une méthode complète qui permet de commander un robot à atteindre un but en évitant des obstacles imprévus en utilisant le moins de moyens possibles en terme de capteurs et de puissance de calcul. Il s'agit aussi de minimiser la période de réglage qui précède l'exécution de la tâche en développant une approche objective qui calcule les paramètres requis de manière automatique.

L'approche développée se base sur la méthode des champs de potentiels artificiels (*APF*) qui consiste à considérer que le but crée une force virtuelle qui attire le robot alors que les obstacles en crée une qui le repousse. Les méthodes basées sur les *APF* sont intéressantes de par leur simplicité mais présentent certains inconvénients qui font entre autre que la trajectoire peut présenter des oscillations indésirables.

Une nouvelle forme du potentiel de répulsion est proposée pour réduire les oscillations et pour éviter les conflits lorsque le but est proche d'un obstacle. Une force additionnelle est aussi proposée qui permet d'améliorer la qualité de la trajectoire dans certains cas d'ambiguïté.

Avec la nouvelle forme des potentiels, une méthode de calcul objectif des paramètres est proposée. Cette méthode fixe des critères d'optimisation et calcule les paramètres en fonction.

La théorie développée est mise à l'épreuve à travers des simulations et des résultats pratiques. Les simulations sont faites sur Matlab dans un environnement contrôlé, alors que les essais pratiques sont effectués sur un robot réel (Pioneer 3AT) dans un environnement moins idéal. Le modèle du robot utilisé est aussi détaillé afin de mettre en contexte les conditions de l'expérimentation.

# **MOBILE ROBOT NAVIGATION IN AN UNKNOWN ENVIRONMENT USING ARTIFICIAL POTENTIAL FIELDS**

Joe Sfeir

## **ABSTRACT**

This project deals with the navigation of a mobile robot in an unknown environment. The aim is to develop a complete method that allows controlling a robot to reach its goal while avoiding unknown obstacles on its way using the least amount of computational or sensing resources. It is also aimed at minimizing the tuning period before the execution of the task at hand, by developing an objective approach that computes the required parameters in an automatic fashion.

The approach developed is based on the Artificial Potential Field (APF) method in which the target creates a virtual force that attracts the robot while obstacles create a virtual force that repels the robot. APF-based methods are interesting because they are simple to implement; however, they have some inherent limitations that create trajectory oscillations, among other things.

A new form of repelling potential is proposed in order to reduce oscillations and to avoid conflicts when the target is close to obstacles. An additional force is proposed as well, allowing for a smoother trajectory around the obstacles.

With the new form of potentials, an objective computation method for the algorithm's parameters is proposed. It uses optimization criteria to determine the adequate value of those parameters.

The theory developed is tested by simulations and experimentation on a real robot. Simulations are done using Matlab in a controlled environment, while practical experimentation is done on a real robot (Pioneer 3AT) in a less than ideal real environment. The robot's model is also detailed in order to put the conditions of the experimentation into context.

## TABLE DES MATIÈRES

	Page
INTRODUCTION	1
CHAPITRE 1 REVUE DE LITTÉRATURE .....	3
1.1. Introduction.....	3
1.2. Génération de trajectoire en environnement connu .....	3
1.2.1. Planification de trajectoire .....	4
1.2.2. Suivi de trajectoire .....	4
1.3. Génération de trajectoire en environnement inconnu .....	5
1.4. Utilisation des champs de potentiels artificiels (APF).....	6
1.5. Synthèse .....	7
CHAPITRE 2 MODÉLISATION DU ROBOT .....	9
2.1. Introduction.....	9
2.2. Différents modèles de robots mobiles – Maalouf (2005) .....	9
2.2.1. Modèle d’Ackermann .....	9
2.2.2. Modèle différentiel à chenilles ou à roues .....	10
2.2.3. Autres modèles.....	11
2.3. Modèle cinématique du robot utilisé.....	13
2.4. Capteurs et mesures .....	16
2.4.1. Capteur proprioceptifs .....	17
2.4.2. Capteurs extéroceptifs.....	22
2.5. Architecture matérielle et logicielle du robot .....	25
2.6. Conclusion .....	29
CHAPITRE 3 NAVIGATION D’UN ROBOT MOBILE.....	30
3.1. Méthode comportementale.....	30
3.2. Méthode des champs de potentiels artificiels .....	32
3.2.1. Caractéristiques et limitations.....	34
3.3. Conclusion .....	40
CHAPITRE 4 CHAMPS DE POTENTIELS ARTIFICIELS MODIFIÉS .....	42
4.1. Introduction.....	42
4.2. Principe .....	42
4.2.1. Distance entre le robot et les obstacles .....	43
4.2.2. Solution au problème <i>GNRON</i> .....	45
4.2.3. Expression de la force de répulsion .....	47
4.2.4. Limitation du potentiel de répulsion .....	50
4.2.5. Minimisation de l’équilibre instable autour des obstacles.....	56
4.3. Point de vue comportemental.....	58
4.4. Conclusion .....	61

CHAPITRE 5	CHOIX OBJECTIF DES PARAMÈTRES .....	62
5.1.	Introduction.....	62
5.2.	Calcul de $\zeta$ et de $\lambda$ .....	62
5.3.	Calcul de $\eta$ et $\eta'$ .....	65
5.4.	Calcul de $m$ .....	70
CHAPITRE 6	RÉSULTATS .....	72
6.1.	Introduction.....	72
6.2.	Critères de comparaison.....	72
6.2.1.	Longueur totale du parcours .....	73
6.2.2.	Durée totale du parcours .....	73
6.2.3.	Coefficient d'oscillations.....	73
6.3.	Simulations .....	74
6.3.1.	Forces de répulsions et de contournement.....	75
6.3.2.	Choix objectif des paramètres.....	86
6.4.	Résultats pratiques .....	94
6.4.1.	Conditions préliminaires.....	94
6.4.2.	Efficacité de la force de contournement .....	96
6.4.3.	Choix objectif des paramètres.....	99
6.5.	Conclusion .....	106
CONCLUSION	.....	107
RECOMMANDATIONS	.....	110
BIBLIOGRAPHIE	.....	111

## LISTE DES FIGURES

	Page
Figure 2-1	Modèle d'Ackermann .....10
Figure 2-2	Modèle différentiel.....11
Figure 2-3	Robot à pattes – Lauron III; Institut de Robòtica I Informàtica Industrial .....12
Figure 2-4	R2K5 - Walking Machine; École de technologie Supérieure .....12
Figure 2-5	Rover - JPL; NASA .....13
Figure 2-6	Robot P3AT .....14
Figure 2-7	Positionnement du robot .....15
Figure 2-8	Encodeur .....17
Figure 2-9	Capteurs de collision.....22
Figure 2-10	Sonars.....23
Figure 2-11	<i>Laser Range Finder</i> .....24
Figure 2-12	Caméra <i>PTZ</i> .....24
Figure 2-13	Architecture de commande .....26
Figure 2-14	ARIA.....27
Figure 2-15	Simulateur <i>MobileSim</i> .....28
Figure 3-1	Méthode comportementale.....31
Figure 3-2	Diagramme comportemental du robot .....32
Figure 3-3	Potentiel parabolique. Les couleurs claires indiquent un potentiel élevé .....35
Figure 3-4	Potentiel conique. Les couleurs claires indiquent un potentiel élevé .....35
Figure 3-5	Potentiel harmonique. Les couleurs claires indiquent un potentiel élevé .....35
Figure 3-6	Allure du potentiel de répulsion d'un obstacle ponctuel .....37
Figure 3-7	Équilibre instable .....39



Figure 4-1	Correction locale du <i>GNRON</i> .....	46
Figure 4-2	Détection d'obstacles.....	48
Figure 4-3	Potentiels de répulsion en fonction de la distance à l'obstacle en grandeurs normalisées. a. potentiel FIRAS - b. forme améliorée du potentiel.....	55
Figure 4-4	Force tangentielle de contournement d'un obstacle .....	57
Figure 4-5	Allure de la fonction de modulation de la force de répulsion.....	59
Figure 4-6	Diagramme de comportements continus.....	60
Figure 6-1	Algorithme de navigation .....	75
Figure 6-2	Forme FIRAS du potentiel de répulsion - $\eta = 1$ .....	77
Figure 6-3	Forme FIRAS du potentiel de répulsion - $\eta = 2$ .....	77
Figure 6-4	Forme FIRAS du potentiel de répulsion - $\eta = 4$ .....	78
Figure 6-5	Nouvelle forme du potentiel de répulsion - $\eta = 1$ .....	79
Figure 6-6	Nouvelle forme du potentiel de répulsion - $\eta = 2$ .....	79
Figure 6-7	Nouvelle forme du potentiel de répulsion - $\eta = 4$ .....	80
Figure 6-8	Nouvelle forme du potentiel de répulsion avec force de contournement - $\eta = 1$	81
Figure 6-9	Nouvelle forme du potentiel de répulsion avec force de contournement - $\eta = 2$ .....	82
Figure 6-10	Nouvelle forme du potentiel de répulsion avec force de contournement - $\eta = 4$ .....	82
Figure 6-11	Forme FIRAS dans le meilleur des cas.....	83
Figure 6-12	Forme améliorée du potentiel de répulsion dans le meilleur des cas.....	84
Figure 6-13	Forme améliorée du potentiel de répulsion avec force de contournement dans le meilleur des cas.....	84
Figure 6-14	Comparaison des performances .....	85
Figure 6-15	Comparaison du comportement du robot avec celui d'un système du second ordre .....	86

Figure 6-16	Distance par rapport à l'origine pour.....	87
Figure 6-17	Variation du paramètre $\lambda$ autour de sa valeur optimale calculée. Valeur trop faible - $\lambda = 0.05$ .....	88
Figure 6-18	Variation du paramètre $\lambda$ autour de sa valeur optimale calculée. Valeur calculée - $\lambda = 0.27$ .....	89
Figure 6-19	Variation du paramètre $\lambda$ autour de sa valeur optimale calculée. Valeur trop grande - $\lambda = 0.9$ .....	89
Figure 6-20	Variation du paramètre $\zeta$ autour de sa valeur optimale calculée. Valeur trop faible - $\zeta = 0.01$ .....	90
Figure 6-21	Variation du paramètre $\zeta$ autour de sa valeur optimale calculée. Valeur calculée - $\zeta = 0.036$ .....	90
Figure 6-22	Variation du paramètre $\zeta$ autour de sa valeur optimale calculée. Valeur trop grande - $\zeta = 0.2$ .....	91
Figure 6-23	Variation du paramètre $\eta$ autour de sa valeur optimale calculée. Valeur trop faible - $\eta = 4$ .....	91
Figure 6-24	Variation du paramètre $\eta$ autour de sa valeur optimale calculée. Valeur calculée - $\eta = 8.44$ .....	92
Figure 6-25	Variation du paramètre $\eta$ autour de sa valeur optimale calculée. Valeur trop élevée - $\eta = 12$ .....	92
Figure 6-26	Valeur optimale de $\lambda$ .....	93
Figure 6-27	Valeur optimale de $\zeta$ .....	93
Figure 6-28	Valeur optimale de $\eta$ .....	94
Figure 6-29	Adaptation de l'algorithme à la cinématique du robot .....	96
Figure 6-30	Forme améliorée du potentiel de répulsion - sans force de contournement ....	97
Figure 6-31	Distance de l'obstacle le plus proche - sans force de contournement .....	97
Figure 6-32	Forme améliorée du potentiel de répulsion - avec force de contournement ....	98
Figure 6-33	Distance de l'obstacle le plus proche - avec force de contournement.....	98

Figure 6-34	Comparaison des performances selon l'utilisation de la force de contournement.....	99
Figure 6-35	Variation du paramètre $\lambda$ autour de sa valeur optimale calculée. Valeur trop faible - $\lambda = 0.1$ .....	100
Figure 6-36	Variation du paramètre $\lambda$ autour de sa valeur optimale calculée. Valeur calculée - $\lambda = 0.27$ .....	100
Figure 6-37	Variation du paramètre $\lambda$ autour de sa valeur optimale calculée. Valeur trop élevée - $\lambda = 0.9$ .....	101
Figure 6-38	Variation du paramètre $\zeta$ autour de sa valeur optimale calculée. Valeur trop faible - $\zeta = 0.01$ .....	101
Figure 6-39	Variation du paramètre $\zeta$ autour de sa valeur optimale calculée. Valeur calculée - $\zeta = 0.036$ .....	102
Figure 6-40	Variation du paramètre $\zeta$ autour de sa valeur optimale calculée. Valeur trop élevée - $\zeta = 0.2$ .....	102
Figure 6-41	Variation du paramètre $\eta$ autour de sa valeur optimale calculée. Valeur trop faible - $\eta = 4$ .....	103
Figure 6-42	Variation du paramètre $\eta$ autour de sa valeur optimale calculée. Valeur calculée - $\eta = 8.44$ .....	103
Figure 6-43	Variation du paramètre $\eta$ autour de sa valeur optimale calculée. Valeur trop élevée - $\eta = 12$ .....	104
Figure 6-44	Valeur optimale de $\lambda$ .....	105
Figure 6-45	Valeur optimale de $\zeta$ .....	105
Figure 6-46	Valeur optimale de $\eta$ .....	106

## LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

$a_m$	Accélération maximale du robot en $m.s^{-2}$
$S$	Abscisse curviligne du robot en $m$
$S_0$	Abscisse curviligne initiale du robot en $m$
$x_0$	Abscisse de la position initiale du robot en $m$
$x_b$	Abscisse du but en $m$
$x$	Abscisse du centre du robot en $m$
$v_x$	Abscisse du vecteur vitesse du robot en $m / s$
<i>GNRON</i>	But Non-Atteignable Lorsque des Obstacles en sont Proches (Goal Non-Reachable when Obstacles are Nearby)
<i>APF</i>	Champs de Potentiels Artificiels (Artificial Potential Fields)
$\zeta$	Coefficient d'attraction du potentiel virtuel du but en $N / m$
$\lambda$	Coefficient de frottement virtuel en $N.s.m^{-1}$
$\eta'$	Coefficient de la force de contournement des obstacles sans unité
$C_o$	Coefficient d'oscillations en $rd / s$
$\eta$	Coefficient de répulsion du potentiel virtuel des obstacles sans unité
$D_a$	Dépassement absolu de la distance entre le robot et l'origine en $m$
$D_{\%}$	Dépassement relatif de la distance entre le robot et l'origine par rapport à la distance entre le but et la position initiale – sans unité
$d_i$	Distance de l'obstacle détecté par le sonar $i$ en $m$
$\rho_0$	Distance de visibilité des sonars en $m$
$\rho$	Distance entre la surface du robot et l'obstacle le plus proche détecté en $m$

$D_b$	Distance entre le centre du robot et le but en $m$
$L$	Distance entre le plan des roues de droite et celui des roues de gauche (Largeur du robot) en $m$
$D_p$	Durée du parcours total en $s$
$\varepsilon_w$	Erreur sur la vitesse de rotation du robot en $rd/s$
$\varepsilon_v$	Erreur sur la vitesse linéaire du robot en $m/s$
$H$	Fonction échelon de Heaviside
$E$	Fonction énergie virtuelle du robot en $J$
$\mu_b$	Fonction modulation de la force d'attraction du but sans unité
$\mu_c$	Fonction modulation de la force de contournement des obstacles sans unité
$\mu_o$	Fonction modulation de la force de répulsion des obstacles sans unité
$FIRAS$	Force Impliquant une Répulsion Artificielle en Surface
$\vec{F}_b$	Force virtuelle d'attraction du but en $N$
$\vec{F}_c$	Force virtuelle de contournement des obstacles en $N$
$\vec{F}_o$	Force virtuelle de répulsion des obstacles en $N$
$L_t$	Longueur totale de la trajectoire du robot en $m$
$m$	Masse virtuelle du robot en $Kg$
$\varepsilon_{m\omega}$	Minimum de l'erreur sur la vitesse de rotation du robot en $rd/s$
$\varepsilon_{mv}$	Minimum de l'erreur sur la vitesse linéaire du robot en $m/s$
$y_0$	Ordonnée de la position initiale du centre du robot en $m$

$y_b$	Ordonnée du but en $m$
$y$	Ordonnée du centre du robot en $m$
$\theta$	Orientation du robot en $rd$
$v_y$	Ordonnée du vecteur vitesse du robot en $m/s$
$\alpha_i$	Position angulaire du sonar $i$
$\phi_b$	Potentiel virtuel d'attraction du but en $J$
$\phi_o$	Potentiel virtuel de répulsion des obstacles en $J$
$\phi_m$	Potentiel virtuel maximal en $J$
$\phi$	Potentiel virtuel total en $J$
$\rho_c$	Rayon de courbure instantané de la trajectoire du robot en $m$
$R$	Rayon du robot (Distance entre le centre et le point le plus éloigné de la surface) en $m$
$T_p$	Temps désiré d'atteinte du but en $s$
$\tilde{\omega}$	Vitesse angulaire réelle du robot en $rd/s$
$V_{d,ar}$	Vitesse de rotation de la roue arrière droite en $rd/s$
$V_{g,ar}$	Vitesse de rotation de la roue arrière gauche en $rd/s$
$V_{d,av}$	Vitesse de rotation de la roue avant droite en $rd/s$
$V_{g,av}$	Vitesse de rotation de la roue avant gauche en $rd/s$
$\omega$	Vitesse de rotation mesurée du robot en $rd/s$
$V_d$	Vitesse des roues de droite en $m/s$
$V_g$	Vitesse des roues de gauche en $m/s$

$V$	Vitesse linéaire mesurée du robot en $m/s$
$\tilde{V}$	Vitesse linéaire réelle du robot en $m/s$
$V_{\max}$	Vitesse maximale du robot en $m/s$

## INTRODUCTION

Les récents développements de la microélectronique, notamment les MEMS et les réseaux de capteurs, semblent mener le domaine de la robotique mobile vers des horizons fort prometteurs. En effet, de plus en plus de compagnies privées ou même des particuliers sont demandeurs d'appareils mobiles autonomes, que ce soit en sécurité, entretien ou même en loisir.

Le développement de la robotique mobile est synonyme de développement des techniques de commande; on cherche à intégrer des techniques de plus en plus performantes sur des structures de plus en plus compactes. Les méthodes de navigation se font variées, et les théories se multiplient et s'approfondissent.

Parmi les techniques de navigation, citons celle des champs de potentiels artificiels. Cette méthode introduite dans les années soixante-dix s'inspire de la théorie des champs électrique et gravitationnel. Elle permet avec une grande simplicité de réaliser la navigation d'un robot dans diverses conditions, notamment sans connaissance à priori de l'environnement.

Dans ce projet on entreprend d'approfondir le développement des champs de potentiels. On applique la méthode développée à un robot mobile afin d'en illustrer les améliorations et limitations.

On présente aussi une méthode qui permet d'initialiser la navigation sans faire intervenir d'opérateur pour faire le réglage; les paramètres de l'algorithme développé sont calculés objectivement selon des critères d'optimisation.

Le premier chapitre commence par rappeler les principales contributions de la littérature scientifique quant au domaine de la navigation. On s'attarde en particulier sur les résultats qui traitent des champs de potentiels artificiels. Les avantages et inconvénients principaux des méthodes vues sont donnés.



Dans le deuxième chapitre, on développe la modélisation du robot utilisé. On voit aussi les capteurs utilisés par le robot. L'architecture matérielle du robot est exposée, ainsi que son architecture logicielle de commande.

Le troisième chapitre traite de la navigation en robotique. Les concepts généraux utilisés dans le projet y sont exposés, ainsi qu'une introduction au concept classique de champs de potentiels. Les limitations de la méthode classique sont expliquées pour introduire les concepts abordés dans le projet.

Dans le quatrième chapitre, on détaille la théorie modifiée des champs de potentiels. Les justifications théoriques sont expliquées ainsi que les détails du développement mathématique.

Le développement de la méthode objective de calcul des paramètres est fait dans le cinquième chapitre. Des critères d'optimisation sont donnés et le développement mathématique des équations est détaillé.

Le dernier chapitre du rapport traite des résultats théoriques et pratiques des essais effectués. L'expérimentation est divisée en cas de simulation et cas pratiques. Les courbes et données obtenues sont discutées et les inconvénients et limitations de la méthode développée dans le projet sont détaillés.

Une conclusion vient présenter une brève synthèse du travail en récapitulant les principaux fondements théoriques derrière la méthode et en donnant les réalisations et les perspectives qui seraient susceptibles d'aboutir à de nouvelles améliorations.

## **CHAPITRE 1**

### **REVUE DE LITTÉRATURE**

#### **1.1. Introduction**

La navigation d'un robot consiste à déterminer une suite de coordonnées que ce dernier doit suivre pour atteindre une destination donnée. On distingue en principe deux types de tâches complémentaires qui permettent au robot de naviguer :

- la génération de trajectoire;
- le suivi de trajectoire.

En effet, ces deux types de tâche doivent être distingués; le suivi de trajectoire permet de commander le robot à passer par des points déterminés, indépendamment de la manière dont ont été générés ces points. La génération de trajectoire consiste en la génération des points de passage, indépendamment de la manière dont le robot s'y rendra.

Lorsque l'environnement du robot est entièrement connu, des méthodes permettent de générer une trajectoire et de contrôler le robot de façon à suivre cette trajectoire. Lorsque l'environnement n'est pas connu à l'avance, il n'est pas possible de séparer la génération de trajectoire du suivi de trajectoire : la trajectoire est générée au fur et à mesure que le robot évolue dans l'environnement.

#### **1.2. Génération de trajectoire en environnement connu**

La génération de trajectoire consiste à interpréter l'ensemble des informations disponibles afin de spécifier une série d'actions à imposer au robot. Ces actions peuvent être exprimées sous plusieurs formes : elles peuvent être un ensemble de points de passage ou une série de vitesses que le robot doit prendre à chaque instant. Les informations dont le robot a besoin pour planifier une trajectoire dépendent directement de l'environnement. La connaissance de l'environnement signifie la connaissance de l'emplacement et de la forme des obstacles.

### **1.2.1. Planification de trajectoire**

Souvent, la planification en environnement connu revient à un problème d'optimisation. Prusky (1996) décrit l'algorithme de Dijkstra qui consiste à diviser la carte de l'environnement en grille et à fixer des points dans les zones accessibles. Un graphe constitué de nœuds et de liens est construit. Le passage entre deux nœuds implique un certain coût (généralement la distance). L'algorithme effectue une recherche exhaustive de façon à minimiser le coût total. La trajectoire finale que le robot doit suivre est un ensemble de segments dont la longueur totale est minimale.

Dans la même optique, d'autres algorithmes sont décrits basés sur le même principe d'optimisation. Prusky (1996) introduit l'algorithme A\* qui utilise une puissance de calcul moins importante que l'algorithme de Dijkstra. Stentz (1994) introduit l'algorithme D\* qui est une combinaison des algorithmes de Dijkstra et A\*. Cet algorithme a l'avantage d'être plus robuste aux incertitudes liées à la position des obstacles.

Citons aussi la génération de trajectoire en utilisant les algorithmes génétiques - Jianping et Yang (2003) - ou les réseaux de neurones – Zegers et Sundareshan (2003).

La génération de trajectoire en environnement connu garantit l'atteinte du but avec une trajectoire optimale selon un certain critère.

### **1.2.2. Suivi de trajectoire**

Une fois la trajectoire générée, il s'agit de faire déplacer le robot de façon à ce qu'il suive cette trajectoire. C'est donc un cas de contrôle à bas niveau qui consiste en une boucle d'asservissement interne à celle de la navigation.

Vue la non-linéarité d'un robot mobile, due à la contrainte non-holonome – Nasrallah (2007), il est nécessaire de concevoir des méthodes de contrôle spécifique.

Des approches basées sur les principes de stabilité et d'énergie consistent à faire converger vers zéro l'erreur entre l'état actuel et l'état désiré du robot en utilisant les principes de stabilité de Lyapunov – Xu et Yang (2001). Koh et Cho (1995) introduisent des méthodes de commande de robot mobile par mode de glissement.

Des méthodes de commande intelligente sont aussi abondantes dans la littérature. Maalouf (2005) montre que l'utilisation de la logique floue permet de commander le robot de façon à passer par les points de passage donnés en utilisant des règles qualitatives. L'utilisation des réseaux de neurones – Yang et Meng (2001) permet de doter le robot de capacités d'apprentissage de façon à pouvoir s'adapter à une grande variété de conditions.

Notons aussi l'utilisation des méthodes basées sur les champs de potentiels. Ces méthodes seront détaillées au paragraphe 1.4.

### **1.3. Génération de trajectoire en environnement inconnu**

Dans plusieurs applications, le planificateur de trajectoire n'a pas nécessairement l'information dont il a besoin pour générer la trajectoire à l'avance. Dans ces conditions, il n'est pas possible de générer une trajectoire complète avant que le robot ne commence à se déplacer; il est nécessaire de générer la trajectoire au fur et à mesure que le robot se déplace tout en recueillant des informations locales sur les obstacles voisins.

Prusky (1996) montre, entre autres, les algorithmes BUG1 et BUG2 qui consistent à faire déplacer le robot droit vers le but, jusqu'à ce qu'il rencontre un obstacle. Un comportement de contournement fait alors déplacer le robot parallèlement à la surface de l'obstacle, jusqu'à ce que le but redevienne accessible. Ce genre d'algorithme est qualifié de comportemental : différents modes d'opération sont programmés séparément et sont déclenchés selon les informations reçues par les différents capteurs du robot.

Yang et Meng (2001) donnent des méthodes basées sur l'utilisation de réseaux de neurones pour générer une trajectoire au fur et à mesure que le robot avance. Zegers et Sundareshan (2003) et Araujo et D'Arbo (1998) utilisent deux réseaux de neurones en cascade; l'un pour planifier la trajectoire pour atteindre le but et l'autre pour faire éviter les obstacles présents sur cette trajectoire. Chang et al. (1994) proposent l'utilisation d'un algorithme génétique pour la génération de la trajectoire globale et l'utilisation d'un réseau de neurones pour les décisions d'évitement d'obstacles.

#### **1.4. Utilisation des champs de potentiels artificiels (APF)**

Introduite dans Khatib (1986), la méthode des champs de potentiels artificiels consiste à attribuer au but un champ de potentiel virtuel attractif. On attribue aux obstacles un champ de potentiel virtuel répulsif. Le potentiel total permet de tirer une force virtuelle qui fait déplacer le robot vers le but en évitant les obstacles.

Plusieurs variantes qui considèrent l'intégralité de l'environnement ont été développées, comme Khasla et Volpe (1998) qui modélisent les obstacles par des superquadriques ou Siemiatkowska (1994) qui utilise les équations de diffusion des fluides pour générer une trajectoire. Ces méthodes sont qualifiées de globales parce qu'elles nécessitent la connaissance de l'environnement dans sa totalité afin de planifier la trajectoire du robot.

Des méthodes locales utilisant les champs de potentiels artificiels sont développées entre autres par Krogh (1984) et Khatib (1986). Ils utilisent la méthode classique des champs de potentiels adaptée à un robot mobile. Cette méthode est simple et ne nécessite pas une très grande puissance de calcul. Elle souffre cependant des inconvénients inhérents aux méthodes de potentiels classiques, notamment le problème des minima locaux et des oscillations.

Le problème des oscillations est amélioré par Ren et al. (2006) qui utilisent une méthode plus élaborée pour le calcul du gradient, ce qui rend la trajectoire plus lisse. Cependant cette

méthode nécessite une puissance de calcul plus grande, ce qui la rend moins accessible à des robots de petite capacité.

Park et Lee (2003) utilisent un potentiel virtuel d'échappement qui s'active lorsque le robot est pris dans un minimum local. Ceci a pour effet de minimiser le risque de ne pas atteindre le but. Cette méthode permet d'améliorer la trajectoire en augmentant les chances d'atteindre le but, mais n'élimine pas le risque d'être pris dans un minimum de potentiel avant d'en échapper. Notons que Park et Lee (2003) introduisent une forme d'utilisation des champs de potentiels qui garantit que le robot ne sera pas pris dans un minimum local, mais elle nécessite la connaissance de l'environnement dans sa totalité.

Le problème GNRON est identifié par Ge et Cui (2000) : le robot ne converge pas exactement vers le but lorsqu'un obstacle se trouve proche de ce dernier. Une nouvelle forme du potentiel de répulsion des obstacles est alors introduite pour garantir que le robot s'immobilise toujours au but.

## **1.5. Synthèse**

Les méthodes locales utilisant les champs de potentiels artificiels font preuve d'une grande simplicité et diversité. Les divers travaux s'y rapportant identifient bien les problèmes et présentent souvent des solutions qui permettent de les pallier.

Cependant, chaque travail présente la solution d'un seul des problèmes à la fois. Aucune approche globale n'a été rencontrée qui présente une méthode complète qui élimine tous les problèmes.

Notons aussi que dans la littérature consultée, les paramètres utilisés dans les différents algorithmes sont choisis sans vraiment donner de méthode fiable pour faire un choix judicieux.

Dans ce travail, on entreprend de synthétiser les méthodes d'amélioration des performances de la méthode des champs de potentiels artificiels ainsi que de présenter une méthode objective qui calcule les valeurs optimales des paramètres utilisés dans l'algorithme.

## **CHAPITRE 2**

### **MODÉLISATION DU ROBOT**

#### **2.1. Introduction**

Dans cette partie, on donne une vue d'ensemble sur différents modèles de robots mobiles. On expose le principe de quelques modèles communs puis on détaille le modèle cinématique du robot à braquage différentiel qui est utilisé dans ce projet. On décrit les capteurs à bord du robot considéré dans notre étude, ainsi que l'architecture de commande du robot.

#### **2.2. Différents modèles de robots mobiles – Maalouf (2005)**

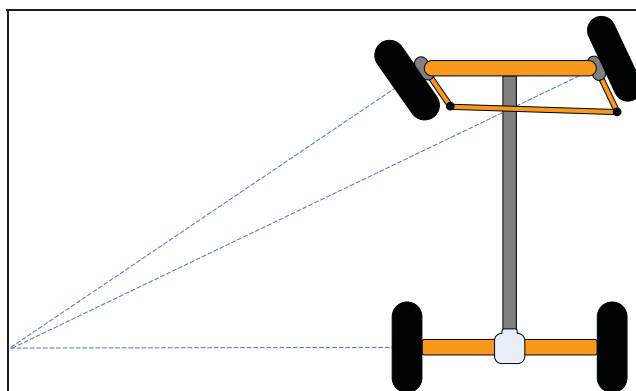
Dépendamment des applications considérées, il existe une grande variété de modèles de robots mobiles utilisables. Il est clair que le cadre de la robotique mobile dépasse largement celui des véhicules à roues et inclut entre autres les véhicules à pattes, les véhicules nautiques, sous-marins et aériens autonomes.

Une caractéristique majeure pour les robots mobiles à roues est la non holonomie : Le déplacement du robot dans le plan ne peut pas se faire dans toutes les directions puisqu'une roue ne peut se déplacer sans glisser que dans son propre plan de rotation. La non holonomie d'un robot est exprimée par une équation liant sa vitesse à son orientation.

##### **2.2.1. Modèle d'Ackermann**

Le modèle d'Ackermann n'est autre que le modèle classique de braquage d'une voiture : le véhicule comporte quatre roues; les deux roues postérieures assurent la traction du véhicule et leurs axes sont alignés alors que les deux roues antérieures assurent le braquage du véhicule et le point de rencontre de leurs axes constitue le centre de rotation instantané du véhicule (Figure 2-1).





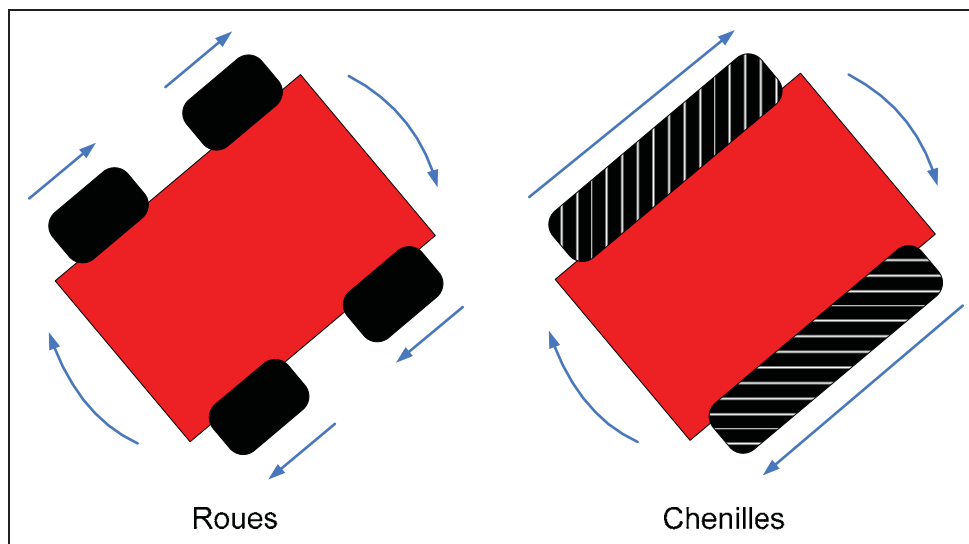
**Figure 2-1 Modèle d'Ackermann**

En effet, vu que les roues du côté extérieur à la courbe suivie sont à une distance plus grande du centre de rotation, elles doivent parcourir une distance légèrement plus grande. Cette différence est assurée par un système mécanique. Le modèle d'Ackermann est généralement utilisé pour sa simplicité. Cependant, un grand inconvénient en est que l'angle de braquage des roues est limité, ce qui fait que le véhicule ne peut parcourir une courbe dont le rayon de courbure est inférieur au rayon de courbure minimal du véhicule. Ceci rend la contrainte non holonome prépondérante lors de la commande en trajectoire de ce type de véhicule.

### **2.2.2. Modèle différentiel à chenilles ou à roues**

Un modèle usuel qui permet d'éliminer le problème du rayon de courbure minimal est le modèle à chenilles. Le véhicule comporte généralement deux chenilles de part et d'autre. La chenille de chaque côté est entraînée par un moteur dédié indépendamment de l'autre chenille. Lorsque les deux côtés roulent à la même vitesse, le véhicule se déplace en ligne droite. Lorsque la vitesse des deux chenilles est différente, le véhicule tourne du côté de la chenille la plus lente. C'est la différence entre ces deux vitesses qui détermine le rayon de courbure qui peut prendre toute valeur désirée. Par contre, ce type de modèle est plus délicat à commander parce qu'il est nécessaire d'asservir la vitesse des deux roues pour permettre au véhicule de se déplacer en ligne droite.

En robotique, les chenilles sont souvent remplacées par des roues; généralement deux roues motrices et une roue folle stabilisatrice. Certains robots ont quatre roues motrices, les deux roues de chaque côté étant entraînées à la même vitesse (Figure 2-2). C'est le cas du robot Pioneer 3AT utilisé dans ce projet.



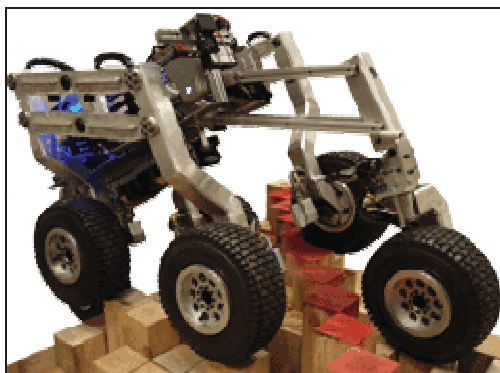
**Figure 2-2 Modèle différentiel**

### 2.2.3. Autres modèles

Dans certains cas, l'application requiert une structure particulière de robot. Souvent, lorsque l'environnement où le robot évolue est accidenté, on utilise un robot à pattes, pour sa capacité à surmonter les accidents de terrain qu'un robot à roues classique ne pourrait pas surmonter (Figure 2-3). Cet avantage est contrecarré par la difficulté du contrôle de ce type de robots. Parfois une structure hybride est utilisée combinant les avantages des structures à pattes et à roues, comme par exemple le robot marcheur *R2K5* du club « *Walking Machine* » de l'ÉTS (Figure 2-4) ou le robot *Rover* du *Jet Propulsion Laboratory* de la *NASA* (Figure 2-5).



**Figure 2-3     Robot à pattes – Lauron III; Institut de Robòtica I Informàtica Industrial**



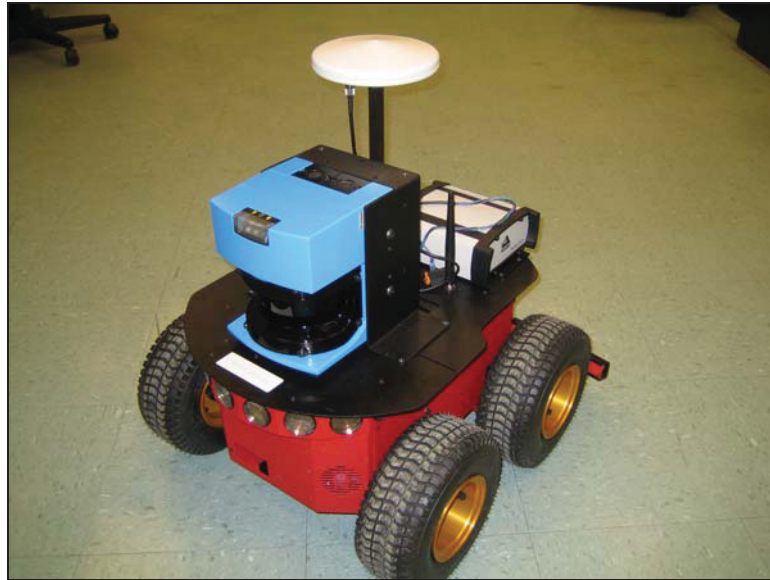
**Figure 2-4     R2K5 - Walking Machine; École de technologie Supérieure**



**Figure 2-5    Rover - JPL; NASA**

### **2.3.    Modèle cinématique du robot utilisé**

On utilise dans ce projet le robot tout-terrain P3-AT développé par MobileRobots. C'est un robot quatre-roues à braquage différentiel du type chenille : les deux roues d'un même côté sont couplées mécaniquement de façon à avoir la même vitesse. Chaque côté est entraîné par un moteur indépendant. On développe ici le modèle cinématique permettant de trouver la position du robot à partir de sa vitesse.



**Figure 2-6 Robot P3AT**

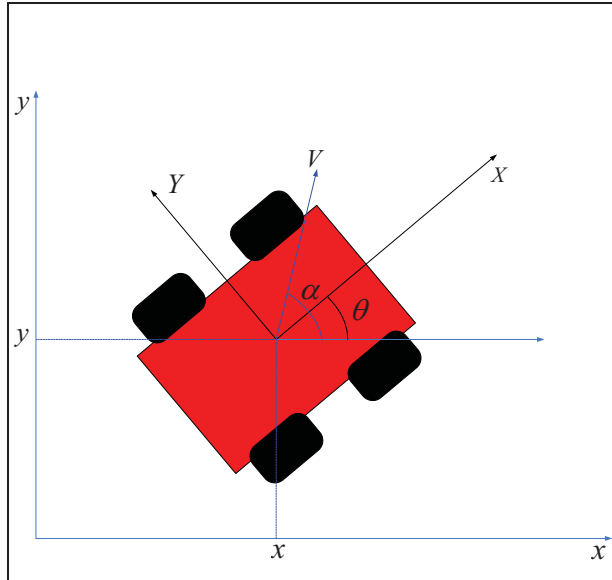
Le positionnement du robot est illustré dans la Figure 2-7.

On commence par définir un repère fixe  $(x, y)$ . Dans ce repère, le centre du robot, défini comme étant le centre du rectangle défini par les centres de roues, a pour coordonnées  $x$  et  $y$ .

On définit aussi le repère  $X, Y$  lié au robot. Le centre du repère est fixé au centre du robot. L'axe  $X$  est défini par l'axe (Avant/Arrière) du robot. L'axe  $Y$  est en quadrature directe par rapport à l'axe  $X$ .

Le positionnement du robot est défini par le vecteur

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad (2.1)$$



**Figure 2-7 Positionnement du robot**

La vitesse du robot est définie par le vecteur

$$\begin{bmatrix} V \\ \omega \end{bmatrix} \quad (2.2)$$

Où  $V$  est la vitesse linéaire du centre du robot et  $\omega$  la vitesse de rotation du robot autour de son centre.

À priori, les coordonnées du vecteur vitesse  $\vec{V}$  du robot dans le repère fixe devraient être

$$\vec{V} = \begin{cases} V \cdot \cos \alpha \\ V \cdot \sin \alpha \end{cases} \quad (2.3)$$

Où  $\alpha$  représente l'angle entre le vecteur  $\vec{V}$  et l'axe  $x$ .

Cependant, la non-holonomie du robot impose que la vitesse du robot soit dans le plan des roues, c'est-à-dire en d'autres termes que  $\alpha = \theta$ .

En effet, on peut affirmer, selon Nasrallah (2007), que

$$V_x \cdot \sin \theta - V_y \cdot \cos \theta = 0 \quad (2.4)$$

Ce qui donnerait

$$\vec{V} = \begin{cases} V \cdot \cos \theta \\ V \cdot \sin \theta \end{cases} \quad (2.5)$$

On peut aussi déduire que la vitesse de rotation du robot autour de son centre est

$$\omega = \dot{\theta} \quad (2.6)$$

Ce qui donne finalement le modèle cinématique du robot

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} V \\ \omega \end{bmatrix} \quad (2.7)$$

Ce même modèle peut être écrit sous la forme d'un système d'équations

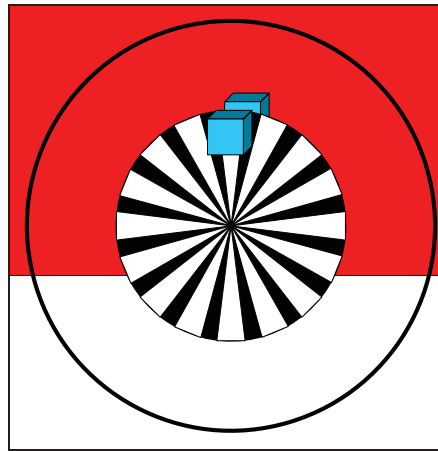
$$\begin{cases} \dot{x} = V \cdot \cos \theta \\ \dot{y} = V \cdot \sin \theta \\ \dot{\theta} = \omega \end{cases} \quad (2.8)$$

## 2.4. Capteurs et mesures

Afin de pouvoir naviguer dans l'environnement, le robot est muni d'un ensemble de capteurs. Des capteurs proprioceptifs lui permettent de calculer son état interne, alors que des capteurs extéroceptifs lui permettent de capter l'état de l'environnement.

### 2.4.1. Capteur proprioceptifs

Les capteurs proprioceptifs principaux sont utilisés pour capter la position du robot. En effet, le robot est muni d'encodeurs de position sur chacune des roues, comme illustré dans la Figure 2-8.



**Figure 2-8 Encodeur**

Les encodeurs calculent la vitesse angulaire des roues et, par intégration numérique, la position angulaire de rotation des roues.

Il est trivial d'affirmer que la vitesse ainsi que la position des deux roues d'un même côté du robot sont égales.

$$\begin{cases} V_{d,av} = V_{d,ar} = V_d \\ V_{g,av} = V_{g,ar} = V_g \end{cases} \quad (2.9)$$

$$\begin{cases} \theta_{d,av} = \theta_{d,ar} = \theta_d \\ \theta_{g,av} = \theta_{g,ar} = \theta_g \end{cases} \quad (2.10)$$

Où  $V_{d,av}$ ,  $V_{d,ar}$ ,  $V_{g,av}$  et  $V_{g,ar}$  représentent respectivement les vitesses de la roue avant droite, arrière droite, avant gauche et arrière gauche.



La connaissance de ces quantités permet de déduire plusieurs grandeurs utiles à la navigation du robot; les vitesses linéaire et angulaire du robot sont calculées selon la formule suivante.

$$\begin{cases} V = \frac{V_d + V_g}{2} \\ \omega = \frac{V_d - V_g}{L} \end{cases} \quad (2.11)$$

Où  $L$  représente la distance entre les centres des roues droite et gauche (avant ou arrière).

Dans certains cas de navigation, il est nécessaire de connaître le rayon de courbure instantanée de la trajectoire du robot. Ce rayon peut être obtenu selon la formule suivante.

$$\rho_c = \frac{V}{\omega} \quad (2.12)$$

En combinant les équations 2.11 et 2.12 on peut déduire :

$$\rho_c = \frac{L}{2} \cdot \frac{V_d + V_g}{V_d - V_g} \quad (2.13)$$

On peut remarquer que, en fonction des vitesses des roues droite et gauche, le rayon de courbure peut prendre toutes les valeurs de  $-\infty$  à  $+\infty$ . Cela met en évidence un atout primordial de cette architecture du robot; il n'y a pas de rayon de courbure minimal imposé, contrairement aux types standard de véhicules avec braquage d'Ackermann.

En mesurant la vitesse des roues, il est donc possible de connaître l'abscisse curviligne  $S$  instantanée du robot par intégration de la relation (2.11) :

$$\begin{cases} S = S_0 + \int_0^t \frac{V_d(t) + V_g(t)}{2} \cdot dt \\ \theta = \theta_0 + \int_0^t \frac{V_d(t) - V_g(t)}{L} \cdot dt \end{cases} \quad (2.14)$$

Réellement, lorsque le robot se déplace, le contact roues-sol n'est pas parfait et il existe un petit glissement tout au long du déplacement du robot. Donc la vitesse réelle du robot n'est pas exactement égale à celle donnée par la formule (2.11), mais plutôt inférieure, puisque lorsque les roues glissent, elles comptabilisent une vitesse de rotation alors que le robot se déplace moins vite; il faut donc y ajouter un facteur d'erreur positif :

$$\begin{cases} \tilde{V} = \frac{V_d + V_g}{2} - \varepsilon_v \\ \tilde{\omega} = \frac{V_d - V_g}{L} - \varepsilon_\omega \end{cases} \quad (2.15)$$

Où  $\varepsilon_v$  et  $\varepsilon_\omega$  représentent l'erreur de calcul due aux glissements non mesurés des roues sur le sol et sont supposés positifs en moyenne.  $\tilde{V}$  et  $\tilde{\omega}$  représentent les vitesses linéaire et angulaire réelles.

La position réelle du robot devrait alors être

$$\begin{cases} \tilde{S} = S_0 + \int_0^t \left( \frac{V_d(t) + V_g(t)}{2} - \varepsilon_v \right) \cdot dt \\ \tilde{\theta} = \theta_0 + \int_0^t \left( \frac{V_d(t) - V_g(t)}{L} - \varepsilon_\omega \right) \cdot dt \end{cases} \quad (2.16)$$

En combinant les équations (2.14), (2.15) et (2.16), on obtient :

$$\begin{cases} \left| \tilde{S} - S \right| = \left| \int_0^t \varepsilon_v \cdot dt \right| \\ \left| \tilde{\theta} - \theta_0 \right| = \left| \int_0^t \varepsilon_\omega \cdot dt \right| \end{cases} \quad (2.17)$$

Vu qu'il est pratiquement impossible de quantifier l'erreur due au glissement puisqu'elle dépend d'une multitude de paramètres dont la vitesse instantanée, la nature du terrain, ainsi que des facteurs stochastiques, on fait une étude plutôt qualitative : si on prend la valeur minimale de l'erreur :

$$\begin{cases} \varepsilon_{mv} = \min(\varepsilon_v) \\ \varepsilon_{m\omega} = \min(\varepsilon_\omega) \end{cases} \quad (2.18)$$

Dans ces conditions, on peut fixer une borne inférieure à l'erreur de position :

$$\begin{cases} \left| \tilde{S} - S \right| \geq \left| \int_0^t \varepsilon_{mv} \cdot dt \right| \\ \left| \tilde{\theta} - \theta_0 \right| \geq \left| \int_0^t \varepsilon_{m\omega} \cdot dt \right| \end{cases} \quad (2.19)$$

En d'autres termes,

$$\begin{cases} \left| \tilde{S} - S \right| \geq \varepsilon_{mv} \cdot t \\ \left| \tilde{\theta} - \theta_0 \right| \geq \varepsilon_{m\omega} \cdot t \end{cases} \quad (2.20)$$

On voit donc que l'erreur entre la position réelle et la position calculée croît avec le temps.

Afin de remédier à ce problème, le robot est muni d'un système de positionnement global (*GPS*) qui permet de mesurer la position du robot sans passer par une intégration de la vitesse des roues. Une antenne de réception ainsi qu'un décodeur sont installés sur le robot pour capter le signal satellite en tout temps.

Ce dispositif permet bien de ne plus passer par une intégration de la vitesse, mais il présente certains désavantages qui ont fait qu'on a dû se limiter au calcul de la vitesse tel qu'expliqué précédemment. En effet, lorsque le robot évolue dans un environnement fermé, le signal *GPS* est trop atténué par les murs et plafonds environnants ce qui fait qu'il est impossible de mesurer la position du robot par *GPS*. Même si le robot se trouve à l'extérieur, dans le cadre de notre application, ses déplacements sont limités à quelques mètres. Or la précision du positionnement global commercial est de quelques mètres, ce qui est trop imprécis pour les petites distances parcourues par le robot dans le cadre des expériences menées. Un meilleur compromis dans le cadre de petits déplacements (six à sept mètres) serait d'utiliser un système de positionnement local par triangulation basé sur des balises fixées à des endroits déterminés dans l'environnement du robot. L'installation d'un tel système nécessite un matériel spécifique et un temps d'installation et de test qui n'étaient pas disponibles. Nous avons donc opté pour le positionnement par intégration de vitesse qui présente un bon compromis précision-distance.

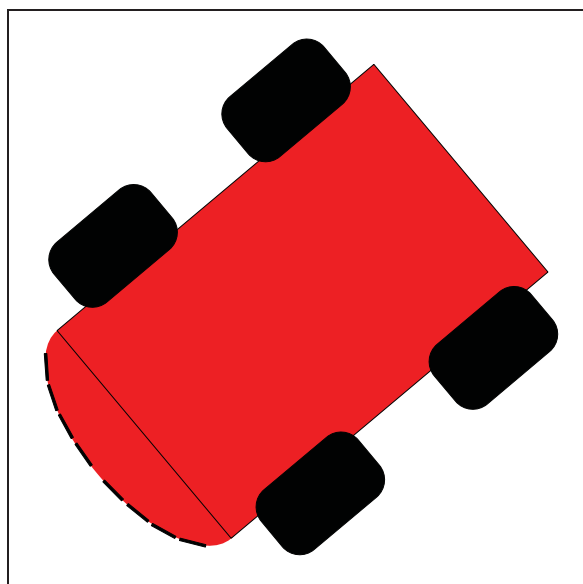
Afin d'améliorer la navigation du robot, il est aussi équipé d'une boussole électronique sensible.

Mis à part les capteurs proprioceptifs cinématiques et inertiels, le robot est équipé d'un nombre de capteurs électriques qui permettent de mesurer différents courants et tensions dans les circuits du robot. Ces capteurs sont utilisés dans une boucle de commande interne qui permet de garantir que les vitesses réelles convergent toujours vers les vitesses de référence désirées. Ces capteurs ne sont pas considérés dans notre étude puisque les boucles de stabilisation internes ne sont pas prises en compte et on ne considère que le système global cinématique.

### 2.4.2. Capteurs extéroceptifs

Afin de capter l'état de son environnement, le robot est muni d'une batterie de capteurs lui donnant différents niveaux de connaissance du milieu.

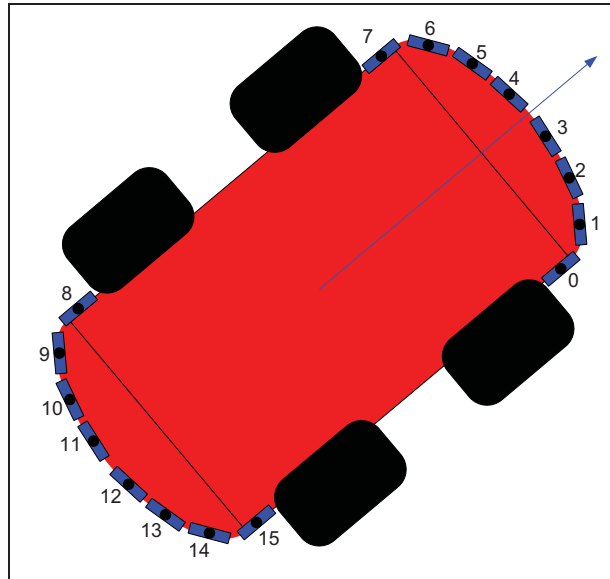
Pour reconnaître une collision avec un obstacle, le robot est muni d'interrupteurs au niveau des pare-chocs, qui se ferment au contact avec un obstacle. Au point de contact, l'interrupteur se ferme et envoie un signal logique VRAI au microprocesseur de contrôle du robot qui traite alors l'événement. Ces capteurs ne sont pas utilisés pour la navigation à proprement parler, mais plutôt comme mesure de sécurité en cas de collision (Figure 2-9). Ils sont placés à l'arrière du robot, puisque d'autres capteurs placés à l'avant peuvent les remplacer pour détecter une collision.



**Figure 2-9 Capteurs de collision**

Tout autour du robot sont disposés 16 capteurs ultrasoniques de distance. Ces sonars envoient un signal sonore haute-fréquence dans une direction déterminée et reçoivent l'onde réfléchiée après un certain temps qui dépend de la distance à un obstacle (Figure 2-10). En fonction de

ce temps, les capteurs retournent au microcontrôleur du robot la distance le séparant de l'obstacle. La distance perceptible par les sonars varie entre 40cm et 5m.



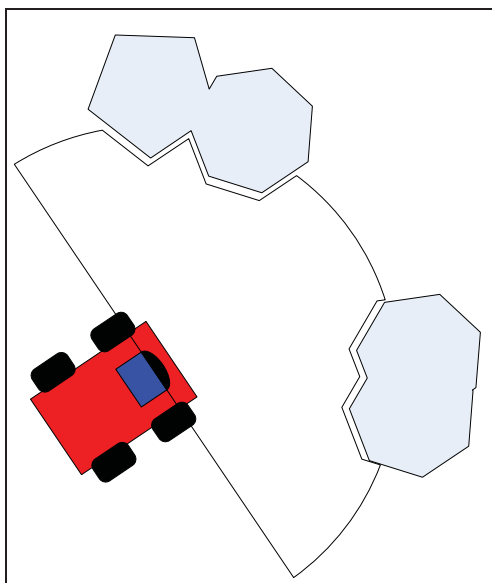
**Figure 2-10 Sonars**

Les sonars sont orientés de la façon suivante par rapport à l'axe.

$$\left\{ \begin{array}{l} \text{Sonar 0 : } -90^\circ \\ \text{Sonar 1 : } -50^\circ \\ \text{Sonar 2 : } -30^\circ \\ \text{Sonar 3 : } -10^\circ \\ \text{Sonar 4 : } 10^\circ \\ \text{Sonar 5 : } 30^\circ \\ \text{Sonar 6 : } 50^\circ \\ \text{Sonar 7 : } 90^\circ \end{array} \right. \quad \left\{ \begin{array}{l} \text{Sonar 8 : } 90^\circ \\ \text{Sonar 9 : } 130^\circ \\ \text{Sonar 10 : } 150^\circ \\ \text{Sonar 11 : } 170^\circ \\ \text{Sonar 12 : } -170^\circ \\ \text{Sonar 13 : } -150^\circ \\ \text{Sonar 14 : } -130^\circ \\ \text{Sonar 15 : } -90^\circ \end{array} \right. \quad (2.21)$$

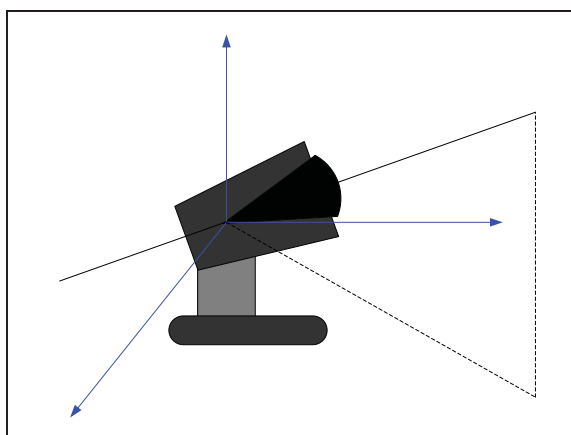
Le robot est aussi équipé d'un *Laser Range Finder* (LRF). Ce dispositif émet un rayon Laser qui oscille à haute-fréquence entre  $-180^\circ$  et  $+180^\circ$  (Figure 2-11). Il permet aussi, par la mesure du rayon réfléchi sur la surface de l'obstacle, de déterminer la position de ce dernier. Ce dispositif a une résolution et une portée bien plus grandes que celles des sonars : au lieu

de donner des distances pour certaines directions bien déterminées, il donne plutôt un profil presque continu de l'environnement devant le robot.



**Figure 2-11** *Laser Range Finder*

Il est aussi possible d'installer une caméra *PTZ* (*Pan-Tilt-Zoom*). Cette caméra permet d'acquérir des images vidéo de l'environnement qui peuvent être utilisés pour plusieurs applications, dont le suivi de routes tracées, suivi d'objets...



**Figure 2-12** *Caméra PTZ*

## 2.5. Architecture matérielle et logicielle du robot

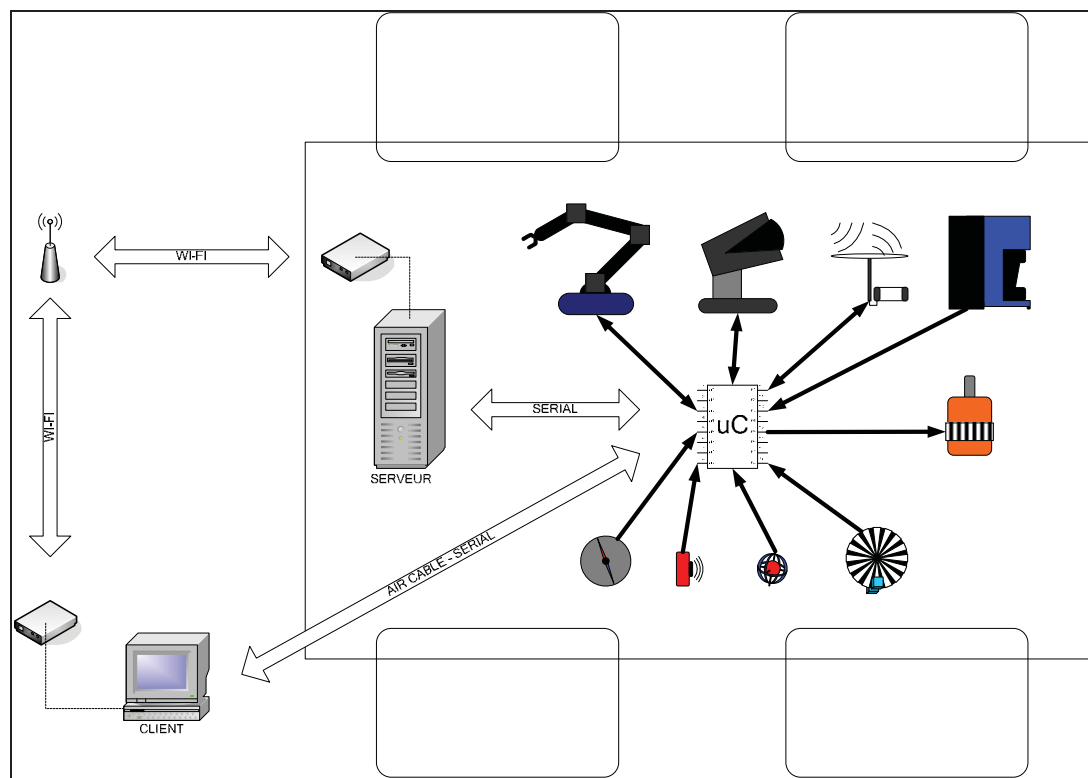
Le robot est mû par quatre moteurs couplés mécaniquement deux à deux. De chaque côté, deux moteurs sont couplés et font tourner les deux roues à la même vitesse à l'aide d'une courroie. Lors du fonctionnement du robot dans un environnement plat, un seul des deux moteurs tourne de chaque côté. Lorsque le robot gravit une pente importante ou doit fournir un couple important, le second moteur de chaque côté est activé. Ceci permet d'optimiser l'utilisation de l'énergie. La gestion de l'activation/désactivation des moteurs d'appoints est faite par le microcontrôleur.

Le robot tire sa puissance d'un groupe de trois batteries de 12V connectées en parallèle. Il est possible d'opérer le robot avec une, deux ou trois de ces batteries installées et les changements peuvent se faire lorsque le robot est actif.

La gestion interne du robot est assurée par un microcontrôleur du type Hitachi H8S. L'alimentation du circuit du microcontrôleur provient du même groupe de batteries mais est maintenue à 5V par un stabilisateur de tension, afin d'éviter les remises à zéro indésirables du microcontrôleur dues aux chutes de tension lorsque les moteurs soutirent un courant important (lors du démarrage ou du blocage du robot). Le microcontrôleur reçoit les signaux provenant des différents capteurs et envoie la commande aux effecteurs. Il peut communiquer aussi avec un ordinateur par communication série RSR232. L'asservissement de la vitesse des moteurs est assuré par le microcontrôleur.

Le robot est également muni d'un ordinateur de bord qui communique avec le microcontrôleur et qui peut être connecté à un réseau via Wi-Fi. Dans notre cas, l'ordinateur de bord n'a pas été utilisé et on s'est connecté directement au microcontrôleur via connexion série sans câble (*Bluetooth*). L'architecture de commande du robot est illustrée sur la Figure 2-13.





**Figure 2-13 Architecture de commande**

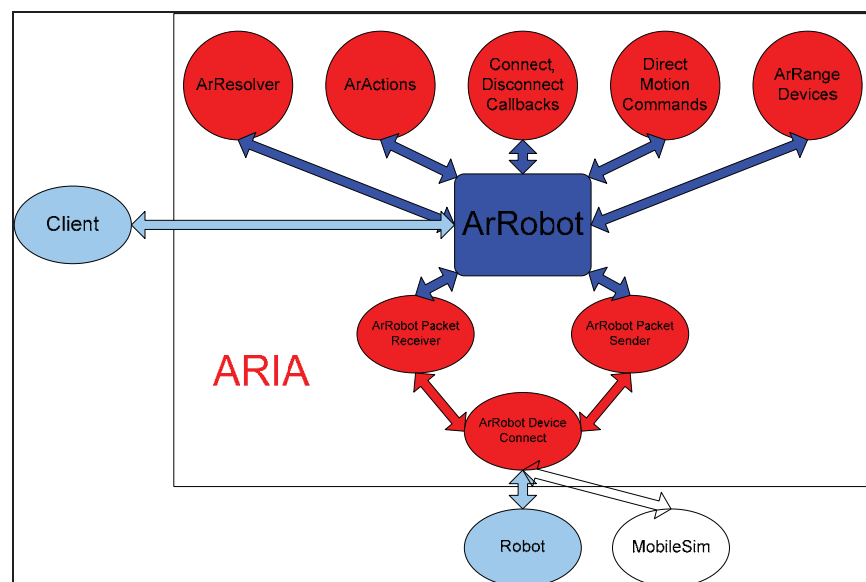
Le robot est construit de façon modulaire et il est possible d'y connecter une grande multitude d'effecteurs et de périphériques, comme un GPS, un bras robotisé, une caméra stéréo, etc.

Puisque le microcontrôleur assure l'asservissement en vitesse du robot, l'utilisateur n'a pas besoin de réaliser la commande bas niveau des moteurs ou des différents périphériques. Le fabricant du robot a mis au point un outil logiciel de commande haut niveau du robot. *Activmedia Robot Interface for Applications (ARIA)* agit comme interface entre l'utilisateur et le robot.

Cet outil consiste en une librairie de classes C++ dont les fonctions servent à envoyer et à recevoir des commandes au microcontrôleur de bord pour commander les actions du robot et recevoir les différentes réponses de ce dernier. Une classe gère une fonctionnalité spécifique du système. Il existe une multitude de classes développées. Les principales sont :

- **ArRobot** : Classe principale qui gère le robot (Avance, orientation, lecture de position, etc.), incluant celles en rapport avec les capteurs;
- **ArRangeDevice** : Gère la communication avec les sonars;
- **ArSimpleConnector** : Gère la communication série avec le microcontrôleur du robot;
- **ArCommands** : Communique directement avec le microcontrôleur en envoyant et recevant des paquets spécifiques d'information;
- **ArAction** : Gère les actions complexes du robot. Ces actions sont programmables par l'utilisateur et sont lancées (Fonction *ArAction::fire*). Les actions permettent de donner des tâches composées au robot, plutôt que de le commander directement.

Avec ARIA, l'utilisateur est capable de se connecter au robot, le conduire à distance, lui envoyer des commandes directes, exécuter un algorithme de navigation et accéder aux périphériques désirés (Figure 2-14).

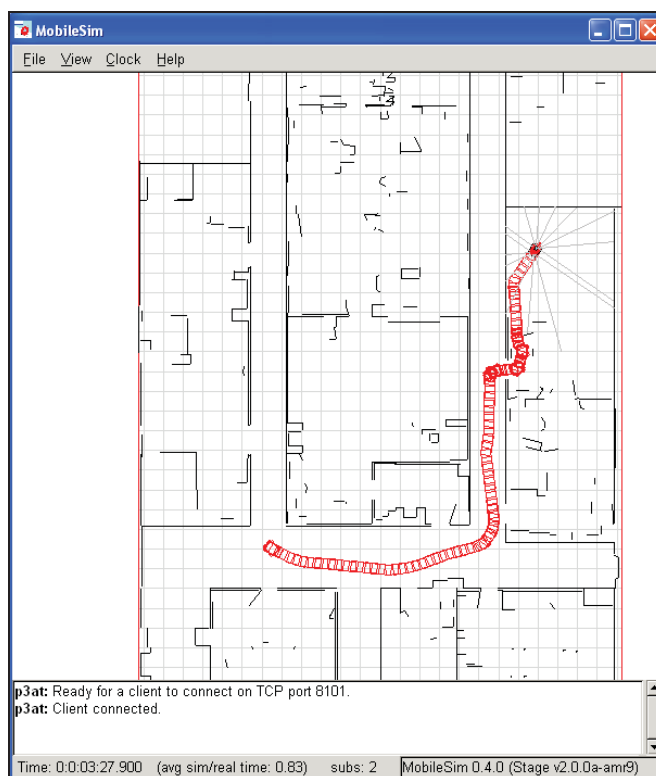


**Figure 2-14 ARIA**

Lorsque le robot n'est pas disponible, il est possible de tester son programme à l'aide d'un simulateur. Le simulateur *MobileSim* (développé par la compagnie *Mobile Robots*) est

utilisable exactement comme le robot : *ARIA* communique avec *MobileSim* et l'utilisateur peut voir l'évolution d'un robot virtuel dans les conditions désirées (Figure 2-15). Il est possible de charger des cartes de l'environnement pour bien modéliser l'évolution du robot. Ces cartes peuvent être construites en format *.txt* ou à l'aide d'un éditeur.

Si désiré, il est même possible d'accéder aux paramètres de la commande bas niveau du microcontrôleur à l'aide d'un outil développé par le fabricant. *Activmedia Robotics Operating System (AROS)* agit comme système d'exploitation du microcontrôleur et l'utilisateur peut, à travers cet outil, configurer les paramètres désirés comme par exemple les facteurs P, I et D des contrôleurs de vitesse des roues.



**Figure 2-15** Simulateur *MobileSim*

## **2.6. Conclusion**

Dans ce chapitre nous avons introduit différents modèles de robots mobiles couramment utilisés. Nous avons développé le modèle cinématique permettant de calculer les quantités principales utilisées dans la navigation du robot. Le robot utilisé pour nos tests a été introduit en détails ainsi que la batterie de capteurs dont il est muni. Les éléments principaux de l'architecture et de la commande du robot ont été introduits. Il aurait été possible de détailler le comportement dynamique du robot, mais cela ne nous est pas utile dans ce cadre, puisque la dynamique du robot est gérée à l'interne et nous ne commandons que le système déjà asservi.

Dans la partie qui suit nous introduisons la méthode des champs de potentiels sur laquelle est basé le travail effectué dans ce projet.

## CHAPITRE 3

### NAVIGATION D'UN ROBOT MOBILE

#### 3.1. Méthode comportementale

Dans cette partie on expose une approche de commande du robot appelée méthode comportementale.

La méthode comportementale consiste à grouper un ensemble de fonctions simples du robot en une fonction de plus haut niveau qui a une signification conceptuelle. L'ensemble de ces fonctions globales de haut niveau constitue le comportement du robot – Prusky (1996).

Par exemple, pour les fonctions de base suivantes :

- $f_1$  : Envoyer une tension  $U$  au moteur gauche;
- $f_2$  : Envoyer une tension  $U$  au moteur droit.

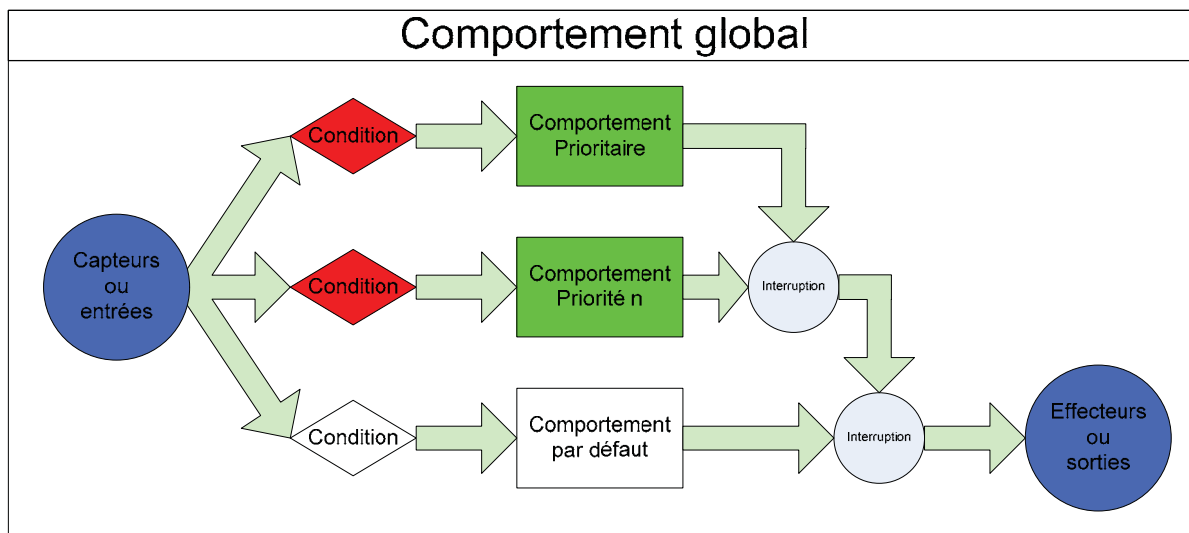
Si on regroupe ces deux fonctions qui, chacune prise séparément, ne font qu'envoyer un signal de tension aux moteurs, le comportement du robot devient un déplacement en ligne droite, ce qui a une signification conceptuelle intelligible par l'utilisateur. Notons que les fonctions constituant un comportement élémentaire peuvent aussi être des comportements elles-mêmes.

Si on ajoute par exemple la condition suivante :

- $g$  : Lorsque le signal d'un sonar indique une distance inférieure à  $50cm$ , Remplacer la tension  $U$  par la tension  $0V$  dans les fonctions  $f_1$  et  $f_2$ .

Cette condition modifie le comportement du robot en désactivant le comportement « déplacement en ligne droite » et en déclenchant le comportement « Arrêt ».

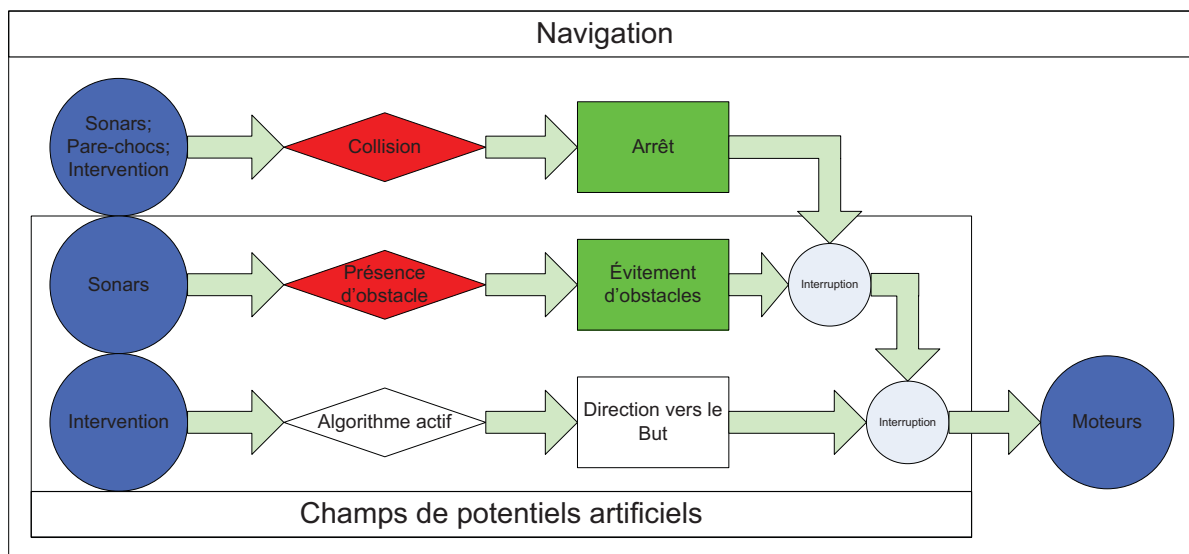
Le comportement global du robot est donc décrit par des comportements individuels qui sont activés et désactivés par certaines conditions : les comportements sont disposés par degrés de priorité, et les conditions interrompent un comportement pour déclencher un comportement de niveau de priorité supérieur (Figure 3-1).



**Figure 3-1 Méthode comportementale**

La complétude et la robustesse du comportement global du robot dépendent de la construction des comportements élémentaires et des conditions d'interruption. La qualité de ces derniers dépend principalement du type, de la diversité, de la qualité et du traitement des capteurs et effecteurs disponibles.

En fonction des capteurs et des effecteurs disponibles pour notre projet, on construit le diagramme de comportements du robot, représenté à la Figure 3-2.



**Figure 3-2 Diagramme comportemental du robot**

Bien entendu, la structure du diagramme comportemental du robot est fonction de l'application désirée. Dans notre cas, le diagramme a été construit de façon à illustrer le plus simplement possible la méthode développée dans ce projet : le comportement global du robot consiste à se rendre à un point déterminé en évitant les obstacles. Finalement, la méthode des champs de potentiels artificiels n'est autre qu'une définition d'un comportement de base qui fait partie du comportement global du robot.

### 3.2. Méthode des champs de potentiels artificiels

Dans cette partie on introduit la théorie des champs de potentiels artificiels (*APF*) telle qu'elle a été développée par Khatib (1986). Le développement de toutes les étapes du calcul est expliqué. Certains inconvénients sont mis en évidence, qui constituent la motivation qui a poussé à développer la méthode modifiée de champs de potentiels qui est la partie centrale du sujet de cette recherche.

Le concept des champs de potentiels artificiels a été introduit initialement pour le contrôle des robots manipulateurs. Le but est de déplacer l'effecteur dans l'espace du robot, tout en

s'assurant que les différents segments du bras n'entrent pas en collision avec les obstacles. La méthode a naturellement été étendue aux robots mobiles.

Il s'agit d'attribuer à l'espace du robot un champ scalaire appelé *champ de potentiel artificiel*. Ce champ est la résultante de deux composantes :

1. Un champ calculé en fonction de la position du but relativement à la position instantanée du robot; ce champ est concave et minimal au but;
2. Un champ calculé en fonction de la forme et de la position relative des obstacles par rapport à celle du robot; ce champ augmente lorsque le robot s'approche d'un obstacle.

D'après Lee (2004) :

$$\phi = \phi_b + \phi_o \quad (3.1)$$

Où  $\phi_b$  représente le champ de potentiel du but,  $\phi_o$  représente celui des obstacles et  $\phi$  représente le champ de potentiel total.

Notons que le champ de potentiel total  $\phi$  est calculé par le contrôleur du robot et n'existe pas physiquement dans l'environnement. Ce n'est qu'un artifice de calcul pour bâtir les fondements théoriques de la méthode. Il permet de calculer une référence de vitesse à donner au robot.

Une fois le champ de potentiel total calculé, un champ de forces est calculé par le gradient du potentiel selon l'équation (3.2) – Lee (2004).

$$\vec{F} = -\overrightarrow{\text{grad}}\phi \quad (3.2)$$

Qui est équivalente à deux équations algébriques :



$$\begin{cases} F_x = -\frac{\partial \phi}{\partial x} \\ F_y = -\frac{\partial \phi}{\partial y} \end{cases} \quad (3.3)$$

Le fait de choisir un champ de forces artificielles qui dérive d'un potentiel revient à dire que ce champ de forces est irrotationnel. Cette supposition est logique puisque les forces virtuelles sont calculées par analogie aux forces de gravitation mécanique. De plus, le fait que le champ de forces soit irrotationnel garantit que le robot ne sera pas pris dans un « vortex », ce qui serait contraire au but de la méthode qui est de le conduire vers le but suivant un chemin relativement direct.

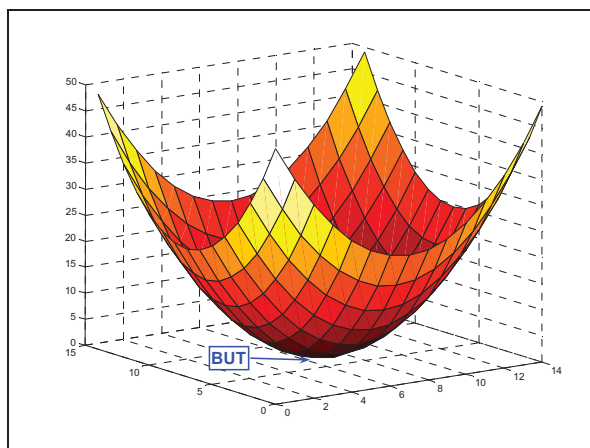
L'expression de la force dans l'équation (3.2) n'est autre que la ligne de plus grande pente, qui rappelle la méthode de descente de gradient classique – Khatib (1986). En d'autres termes le robot se déplace vers le but suivant les potentiels décroissants, dans la direction où la chute de potentiel est la plus grande.

La force calculée en chaque point permet de déduire une vitesse de référence à imposer au robot. Dans le cas classique, la vitesse de référence est confondue avec la force :

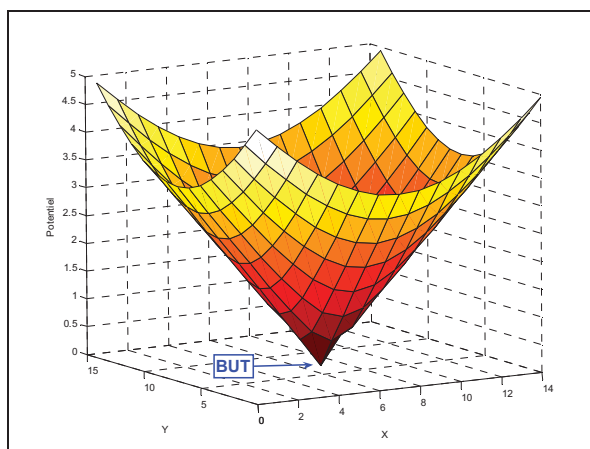
$$\begin{cases} v_x = F_x = -\frac{\partial \phi}{\partial x} \\ v_y = F_y = -\frac{\partial \phi}{\partial y} \end{cases} \quad (3.4)$$

### 3.2.1. Caractéristiques et limitations

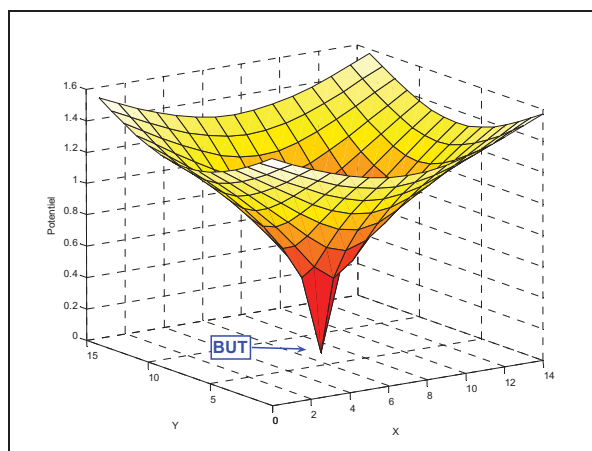
Revenons au calcul du potentiel selon l'équation (3.1). La forme du potentiel d'attraction du but  $\phi_b$  est concave et minimale au but. L'allure d'un tel potentiel peut être selon la Figure 3-3, la Figure 3-4 ou la Figure 3-5.



**Figure 3-3 Potentiel parabolique. Les couleurs claires indiquent un potentiel élevé**



**Figure 3-4 Potentiel conique. Les couleurs claires indiquent un potentiel élevé**



**Figure 3-5 Potentiel harmonique. Les couleurs claires indiquent un potentiel élevé**

Dans le cas classique, le potentiel du but utilisé est parabolique (Figure 3-3) – Khatib (1986) :

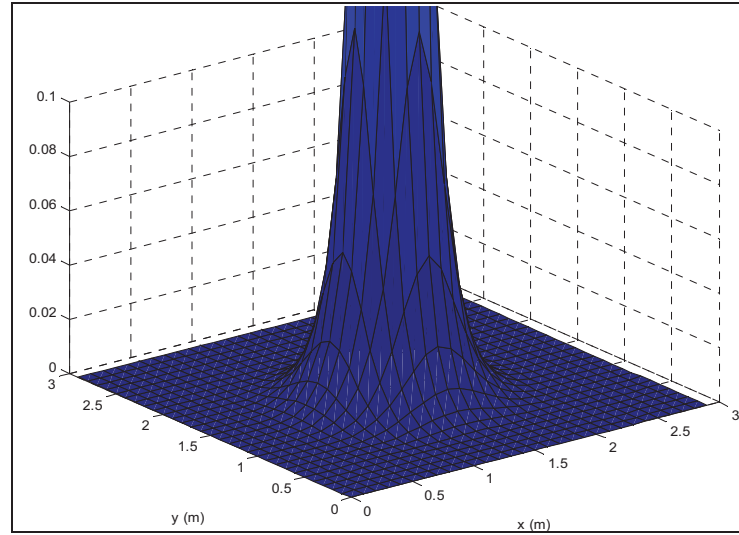
$$\phi_b = \frac{1}{2} \cdot \zeta \cdot \left[ (x - x_b)^2 + (y - y_b)^2 \right] \quad (3.5)$$

Où  $\zeta$  représente le coefficient d'attraction du but et  $x_b$  et  $y_b$  représentent les coordonnées du but.

La forme usuelle du potentiel répulsif des obstacles est donnée par la fonction *FIRAS* - Lee (2004) :

$$\begin{cases} \phi_o = \frac{1}{2} \cdot \eta \cdot \left( \frac{1}{\rho} - \frac{1}{\rho_0} \right)^2 ; \rho \leq \rho_0 \\ \phi_o = 0 ; \rho > \rho_0 \end{cases} \quad (3.6)$$

Où  $\rho$  représente la plus petite distance entre le robot et la surface de l'obstacle considéré et  $\rho_0$  est une constante caractérisant le rayon de sensibilité du robot aux obstacles (dans le cas de l'utilisation de sonars, cette constante n'est autre que la distance de visibilité des sonars).  $\eta$  est un facteur multiplicatif constant. La forme du potentiel de répulsion d'un obstacle a une allure hyperbolique (Figure 3-6).



**Figure 3-6 Allure du potentiel de répulsion d'un obstacle ponctuel**

Pour la force on obtient :

$$\begin{cases} F_{xb} = -\zeta(x - x_b) \\ F_{yb} = -\zeta(y - y_b) \end{cases} \quad (3.7)$$

$$\begin{cases} F_{xo} = \eta \cdot \frac{1}{\rho^2} \cdot \left( \frac{1}{\rho} - \frac{1}{\rho_0} \right) \cdot \frac{\partial \rho}{\partial x}; \rho \leq \rho_0 \\ F_{yo} = \eta \cdot \frac{1}{\rho^2} \cdot \left( \frac{1}{\rho} - \frac{1}{\rho_0} \right) \cdot \frac{\partial \rho}{\partial y}; \rho \leq \rho_0 \\ F_{xo} = 0; \rho > \rho_0 \\ F_{yo} = 0; \rho > \rho_0 \end{cases} \quad (3.8)$$

Une fois la force calculée, il s'agit de déduire la vitesse de référence à donner au robot. Bien qu'il soit simple de choisir la vitesse selon la relation (3.4), ce choix ferait que le robot oscillerait autour du but sans jamais se stabiliser. Un choix commun consiste à mener l'analogie mécanique un pas plus loin en considérant une dynamique virtuelle du robot et en introduisant une force de frottement fluide virtuelle. On obtient :

$$\vec{F} - \lambda \cdot \vec{v} = m \cdot \frac{d\vec{v}}{dt} \quad (3.9)$$

Où  $\vec{F}$  représente la force virtuelle appliquée au robot,  $\vec{v}$  la vitesse de référence à donner au robot et  $m$  la masse virtuelle du robot qui n'est pas nécessairement égale à la masse réelle du robot. Le coefficient  $\lambda$  est un facteur de frottement.

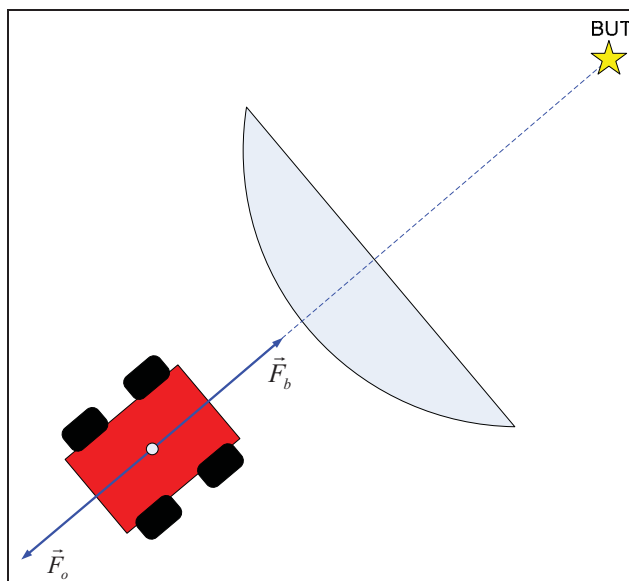
Naturellement, cette analogie avec la méthode de descente de gradient rappelle les inconvénients de cette dernière - Bertsekas (1999).

En effet, la méthode classique de descente de gradient donne au système un comportement dont les caractéristiques sont similaires à celles d'un système oscillatoire du second ordre, comme nous le verrons au CHAPITRE 4. Un tel système est intrinsèquement oscillatoire. La trajectoire générée par une telle méthode a donc facilement tendance à osciller. Une des propriétés inhérentes à la méthode des champs de potentiels classique est l'oscillation de la trajectoire lorsque le robot se trouve dans des passages étroits – Ren et al. (2006).

De par la similarité avec la descente de gradient, il est aussi clair que la méthode de champs de potentiels classique est fort vulnérable aux puits de potentiel, soit les minima locaux qui piègent le robot. Il se trouve alors dans une position d'équilibre stable, s'immobilise et ne peut plus atteindre le but. Ce problème survient souvent à proximité d'un obstacle dont la surface est concave, ou lorsque les obstacles sont proches les uns des autres Lee (2004).

Un autre problème, qui est en fait de même nature que celui des minima locaux est, dans la plupart des cas, lorsque la trajectoire locale du robot est perpendiculaire à la surface d'un obstacle (Figure 3-7). Ce cas limite met le robot dans une position d'équilibre instable. Dans ce cas le robot n'est pas vraiment pris dans un minimum local; cela ne l'empêche pas d'atteindre le but, parce que pratiquement il y a toujours une perturbation qui va faire que sa direction ne sera pas parfaitement perpendiculaire à la surface locale de l'obstacle. Cependant, cette perturbation est souvent faible par rapport aux forces virtuelles mises en

jeu, ce qui fait qu'il faut beaucoup de temps au robot pour contourner l'obstacle. Cela se traduit par des oscillations locales qui diminuent au fur et à mesure que le robot contourne l'obstacle et s'éloigne de cette position singulière. Nous verrons dans le CHAPITRE 6 les résultats d'essais dans ces conditions.



**Figure 3-7 Équilibre instable**

En revenant à l'expression du potentiel (Équations 3.5 et 3.6), une autre limitation peut être mise en évidence. En effet, l'expression du potentiel du but fait bien en sorte qu'il est minimal lorsque le robot se trouve au but. Cependant, l'expression du potentiel de répulsion des obstacles ne l'est pas; lorsqu'un obstacle se trouve suffisamment proche du but, son potentiel n'y est pas nul et le potentiel total, étant la somme de ces deux composantes, n'est pas minimal au but. Dans ces conditions, le robot n'atteindra pas exactement le but mais se placera à des coordonnées différentes.

Ce problème est désigné par *GNRON* (*Goal Not Reachable when Obstacles are Nearby*). Il a été identifié par Ge et Cui (2000) et une solution a été proposée : pour garantir que le potentiel total est minimal au but, il suffit de faire en sorte que le potentiel de répulsion des obstacles le soit. Pour cela, ils proposent une forme modifiée du potentiel de répulsion :

$$\begin{cases} \phi_o = \frac{1}{2} \cdot \eta \cdot \left( \frac{1}{d} - \frac{1}{\rho_0} \right)^2 \cdot D_b^n & ; \rho \leq \rho_0 \\ \phi_o = 0 & ; \rho > \rho_0 \end{cases} \quad (3.10)$$

Où  $d$  représente la distance du robot à l'obstacle,  $D_b$  représente la distance du robot au but et  $n$  est un paramètre ajustable.

Cette forme du potentiel des obstacles permet bien d'éviter le problème *GNRON*. Cependant la nouvelle équation déforme le potentiel dans tout l'environnement uniquement pour corriger un problème local. Ceci n'est pas toujours un problème majeur, mais nous proposons dans ce travail (CHAPITRE 4) une forme modifiée qui permet d'éviter cette distorsion globale.

Un autre point notable est celui du choix des paramètres de l'algorithme. En effet, nous avons vu dans l'expression des potentiels et dans la dynamique virtuelle du robot certains paramètres constants :

$$\eta; \zeta; \lambda; m \quad (3.11)$$

Le choix de ces paramètres est crucial pour la performance de la méthode. Nous verrons dans le CHAPITRE 5 une méthode de choix objectif de ces paramètres selon certains critères d'optimisation.

### 3.3. Conclusion

Nous avons vu dans ce chapitre le principe de l'approche comportementale pour la navigation. La méthode des champs de potentiels artificiels a été introduite telle que traitée dans le cas classique. Les limitations de cette dernière ont été identifiées et expliquées.

Dans le prochain chapitre nous entreprenons de développer une méthode ajustée des champs de potentiels qui permettent de résoudre les problèmes inhérents de la méthode classique.



## **CHAPITRE 4**

### **CHAMPS DE POTENTIELS ARTIFICIELS MODIFIÉS**

#### **4.1. Introduction**

Dans ce chapitre on adapte la méthode des champs de potentiels artificiels à notre application. Les conditions et restrictions sont énumérées. On détaille les étapes de construction de l'algorithme.

On commence par expliquer le principe de la méthode développée en partant des limitations énumérées au chapitre précédent. On passe ensuite à l'explication détaillée des hypothèses et des développements effectués.

#### **4.2. Principe**

D'abord, commençons par citer le cadre de la méthode : on cherche à naviguer le robot de sa position initiale vers le but en évitant les obstacles. Les conditions que l'on se fixe sont les suivantes :

- le robot n'a pas d'information globale sur l'environnement; aucune carte de l'environnement n'est mise en mémoire et aucune informations préalable n'est connue concernant les obstacles;
- le robot n'a comme entrées que celles de ses capteurs (Voir CHAPITRE 2);
- aucune trajectoire à suivre n'est définie. La méthode consiste à générer une trajectoire en temps réel.

Notons qu'aucune condition à priori n'est posée concernant le comportement des obstacles : puisque ceux là sont inconnus, il est intéressant de ne pas en imposer l'immobilité. La validité de cette hypothèse sera testée par l'expérimentation au CHAPITRE 6.

Nous avons vu dans le chapitre précédent le principe de base de la méthode des champs de potentiels. Les limitations principales de la méthode nous ont poussés à développer une approche pour contourner chacune de ces limitations.

#### 4.2.1. Distance entre le robot et les obstacles

Revenons au potentiel artificiel de répulsion des obstacles décrit dans l'équation (3.6) (Réécrite ici pour commodité) :

$$\begin{cases} \phi_o = \frac{1}{2} \cdot \eta \cdot \left( \frac{1}{\rho} - \frac{1}{\rho_0} \right)^2 ; \rho \leq \rho_0 \\ \phi_o = 0 ; \rho > \rho_0 \end{cases} \quad (4.1)$$

Il s'agit de définir la distance  $\rho$  dans les conditions que nous nous sommes fixées. En effet, une carte de l'environnement n'est pas disponible et le calcul de cette distance doit se faire uniquement en fonction des données des capteurs du robot.

Les capteurs extéroceptifs pertinents pour notre robot sont le *Laser Range Finder* et les sonars. À priori il est intéressant d'utiliser le plus de capteurs possible afin d'avoir une redondance qui permet de réduire l'erreur de mesure. Cependant, dans le souci de développer la méthode la plus portable possible, nous avons décidé de commencer par se limiter aux sonars. De cette façon notre méthode serait applicable à pratiquement tout robot mobile puisque les sonars sont les capteurs les plus utilisés. Les résultats expérimentaux du CHAPITRE 6 valident cette hypothèse.

Avec les sonars installés sur le robot, l'information recueillie sur les obstacles avoisinants est un ensemble de seize distances  $d_i$  comprises chacune entre 0 et  $\rho_0$ ,  $\rho_0$  étant la distance maximale de détection des sonars.

On cherche donc la distance  $\rho$  en fonction de ces seize distances :

$$\rho = \rho(d_i)_{0 \leq i \leq 15} \quad (4.2)$$

Le premier choix intuitif serait d'affirmer que la distance  $\rho$  devrait être choisie comme la moyenne arithmétique des distances détectées par les sonars :

$$\rho = \frac{\sum_{i=0}^{15} d_i}{16} \quad (4.3)$$

Cependant il est important de rappeler une condition implicite de réussite de l'algorithme : il faut garantir que le robot n'entrera pas en collision avec un obstacle. En d'autres termes, aucune des distances  $d_i$  ne devrait atteindre 0. En utilisant la moyenne arithmétique, on a :

$$\min(d_i) \leq \rho \leq \max(d_i) \quad (4.4)$$

Donc si on garantit que  $\rho$  est toujours non nul, cela n'implique pas forcément que la distance minimale  $\min(d_i)$  ne s'annulerait pas.

Il serait donc plus intéressant de choisir par exemple :

$$\rho = \min(d_i) \quad (4.5)$$

Ce choix garantirait le contrôle sur la distance minimale et éviterait une collision. Nous avons développé la méthode autour de ce choix et avons obtenu des résultats intéressants.

Donc une expression adaptée du potentiel de répulsion des obstacles serait :

$$\phi_o = \frac{1}{2} \cdot \eta \cdot \left( \frac{1}{\min(d_i)} - \frac{1}{\rho_0} \right) \quad (4.6)$$

#### 4.2.2. Solution au problème *GNRON*

Cette forme du potentiel des obstacles est bien adaptée à notre cas. Cependant, le problème du *GNRON* identifié précédemment est encore présent.

Selon Ge et Cui (2000), il est possible de résoudre le problème en multipliant le potentiel d'attraction par une quantité qui s'annule au but. La forme proposée est citée dans l'équation (3.10) et reprise ici pour commodité :

$$\begin{cases} \phi_o = \frac{1}{2} \cdot \eta \cdot \left( \frac{1}{d} - \frac{1}{\rho_0} \right)^2 \cdot D_b^n & ; \rho \leq \rho_0 \\ \phi_o = 0 & ; \rho > \rho_0 \end{cases} \quad (4.7)$$

Dans cette équation, le problème du *GNRON* est bien résolu, cependant la quantité qui multiplie le potentiel crée une distorsion du potentiel de répulsion du but en tout point; cette distorsion est d'autant plus forte qu'on s'éloigne du but, ce qui est contre-intuitif puisque c'est uniquement aux alentours du but que le problème existe.

On propose donc une quantité multiplicative qui s'annule bien au but, qui est strictement positive partout ailleurs, mais qui est négligeable quand le robot est relativement loin du but. De cette façon la distorsion est locale et n'affecte pas le comportement du robot lorsqu'il est loin du but :

$$\phi_o = \frac{1}{2} \cdot \eta \cdot \left( \frac{1}{\min(d_i)} - \frac{1}{\rho_0} \right) \cdot \left( 1 - e^{-\frac{D_b^2}{R^2}} \right) \quad (4.8)$$

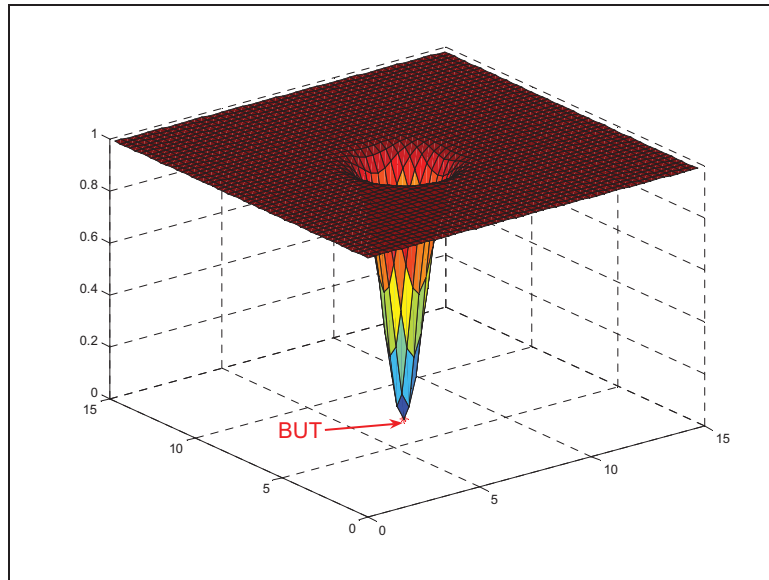
Où

$$D_b = (x - x_b)^2 + (y - y_b)^2 \quad (4.9)$$

Représente la distance entre le centre du robot et le but et  $R$  représente le rayon du robot.

Notons que la quantité  $\min(d_i)$  n'est nulle que lorsque le robot est en contact avec un obstacle.

La quantité multiplicative  $1 - e^{-\frac{D_b^2}{R^2}}$  est une gaussienne inversée qui ne modifie pas le potentiel lorsque le robot s'éloigne du but (Figure 4-1).



**Figure 4-1** Correction locale du *GNRON*

On peut considérer que le rayon du robot agit comme l'écart-type de la gaussienne et donc la distorsion devient négligeable lorsque le robot se trouve à plus de  $3R$  du but. En d'autres termes, lorsque le robot n'est pas proche du but, on peut négliger la présence du facteur multiplicatif. Rappelons que cette modification du potentiel de répulsion des obstacles doit supposer que le but est suffisamment loin des obstacles pour que le robot puisse s'y placer sans toucher d'obstacle. Cette hypothèse est légitime et est adoptée dans tout le projet.

### 4.2.3. Expression de la force de répulsion

Une autre modification apportée à l'expression du potentiel de répulsion est motivée par le développement d'une méthode objective de calcul des paramètres (CHAPITRE 5). En revenant à l'expression du potentiel de répulsion des obstacles (4.8), on voit que lorsque  $\min(d_i)$  s'approche de 0, le potentiel de répulsion  $\phi_o$  tend vers l'infini. Ceci garantit bien l'évitement de l'obstacle. Cependant, en revenant à l'expression de la force de répulsion dans le cas où le robot est loin du but, c'est-à-dire en négligeant la présence du facteur multiplicatif, on obtient l'expression de l'équation (3.8) reprise ici pour commodité :

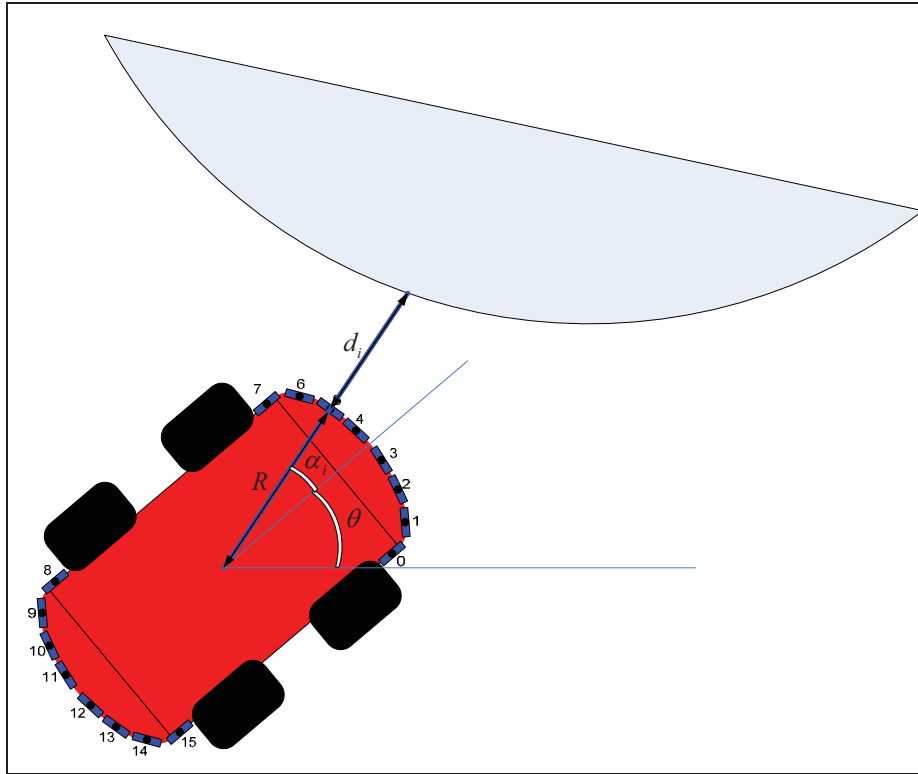
$$\begin{cases} F_{xo} = \eta \cdot \frac{1}{\rho^2} \cdot \left( \frac{1}{\rho} - \frac{1}{\rho_0} \right) \cdot \frac{\partial \rho}{\partial x} \\ F_{yo} = \eta \cdot \frac{1}{\rho^2} \cdot \left( \frac{1}{\rho} - \frac{1}{\rho_0} \right) \cdot \frac{\partial \rho}{\partial y} \end{cases} \quad (4.10)$$

On tente maintenant de détailler l'expression des dérivées  $\frac{\partial \rho}{\partial x}$  et  $\frac{\partial \rho}{\partial y}$ .

En désignant par  $x_{oi}$  et  $y_{oi}$  les coordonnées de l'obstacle détecté par le sonar  $i$ , nous avons (Figure 4-2) :

$$(x_{oi} - x)^2 + (y_{oi} - y)^2 = (R + d_i)^2 \quad (4.11)$$

Où  $R$  représente la distance entre le sonar et le centre de gravité du robot puisque les sonars se trouvent sur la périphérie du robot.  $\alpha_i$  représente l'orientation du sonar  $i$  par rapport à celle du robot et  $\theta$  est l'orientation du robot. Notons que dans ce cas, on suppose que le sonar est orienté suivant le rayon du robot. En réalité ce n'est pas le cas, mais les fonctions utilisées avec *ARIA* retournent des coordonnées qui peuvent être transformées pour donner des positions équivalentes des sonars qui vérifient cette condition.



**Figure 4-2** Détection d'obstacles

On peut alors écrire :

$$d_i = \sqrt{(x_{oi} - x)^2 + (y_{oi} - y)^2} - R \quad (4.12)$$

En dérivant cette expression par rapport aux coordonnées du robot, on obtient :

$$\begin{cases} \frac{\partial d_i}{\partial x} = \frac{x - x_{oi}}{\sqrt{(x_{oi} - x)^2 + (y_{oi} - y)^2}} \\ \frac{\partial d_i}{\partial y} = \frac{y - y_{oi}}{\sqrt{(x_{oi} - x)^2 + (y_{oi} - y)^2}} \end{cases} \quad (4.13)$$

Or nous avons :

$$\begin{cases} x_{oi} = x + (R + d_i) \cos(\theta + \alpha_i) \\ y_{oi} = y + (R + d_i) \sin(\theta + \alpha_i) \end{cases} \quad (4.14)$$

D'où les expressions recherchées :

$$\begin{cases} \frac{\partial d_i}{\partial x} = -\cos(\theta + \alpha_i) \\ \frac{\partial d_i}{\partial y} = -\sin(\theta + \alpha_i) \end{cases} \quad (4.15)$$

En réécrivant la force on obtient :

$$\begin{cases} F_{xo} = -\eta \cdot \frac{1}{\rho^2} \cdot \left( \frac{1}{\rho} - \frac{1}{\rho_0} \right) \cdot \cos(\theta + \alpha_i) \\ F_{yo} = -\eta \cdot \frac{1}{\rho^2} \cdot \left( \frac{1}{\rho} - \frac{1}{\rho_0} \right) \cdot \sin(\theta + \alpha_i) \end{cases} \quad (4.16)$$

Le module de la force de répulsion est donc :

$$F_o = \sqrt{F_{xo}^2 + F_{yo}^2} = \eta \cdot \frac{1}{\rho^2} \cdot \left( \frac{1}{\rho} - \frac{1}{\rho_0} \right) \quad (4.17)$$

Avec cette expression de la force, on peut voir que lorsque  $\rho$  tend vers 0, c'est-à-dire lorsque le robot s'approche indéfiniment de l'obstacle, la force de répulsion tend vers l'infini comme l'inverse du cube de la distance. Ce comportement garantit bien l'évitement de l'obstacle mais peut devenir trop conservateur. En effet, dans ces conditions on ne peut pas avoir de contrôle sur l'intensité de la force lorsque le robot s'approche du robot. En revenant à la dynamique virtuelle du robot, on perd le contrôle sur l'accélération virtuelle qui peut devenir



arbitrairement grande. Comme nous le verrons dans (CHAPITRE 5), il est préférable de garder l'accélération à l'intérieur d'une limite déterminée.

Nous cherchons maintenant une expression du potentiel qui soit moins conservatrice mais qui garantit quand même l'évitement des obstacles quelles que soient les conditions.

#### 4.2.4. Limitation du potentiel de répulsion

Afin de mieux comprendre le comportement du robot lorsqu'il est soumis au potentiel artificiel et lorsqu'il suit la dynamique introduite au CHAPITRE 3, introduisons la fonction  $E$  :

$$E = 2\phi + \frac{1}{2}m\dot{x}^2 + \frac{1}{2}m\dot{y}^2 \quad (4.18)$$

Où  $\phi$  représente le potentiel total auquel est soumis le robot,  $\dot{x}$  et  $\dot{y}$  représentent les coordonnées de la vitesse du robot dans le repère fixe et  $m$  représente la masse virtuelle du robot. Afin de simplifier les calculs, et sans perdre la généralité du résultat, il est possible d'effectuer une translation du repère fixe de façon à ce que le but soit placé à l'origine du repère.

Par analogie à un système mécanique,  $E$  représente l'énergie virtuelle du système robot-but-obstacles.

On cherche à prouver que l'énergie  $E$  est une fonction de Lyapunov.

On sait que lorsque le robot se trouve au but, dans les conditions où le problème *GNRON* est éliminé, le potentiel du robot au but est nul :

$$\phi(0,0) = 0 \quad (4.19)$$

De plus, lorsque le robot atteint le but, la force totale est nulle et il s'immobilise; d'où :

$$\dot{x} = \dot{y} = 0 \quad (4.20)$$

Et donc l'énergie totale du système est nulle au but :

$$E(0,0) = 0 \quad (4.21)$$

De plus, d'après l'expression du potentiel total du robot, on peut affirmer que  $\phi > 0$  et donc

$$E > 0; \forall (x, y) \neq (0,0) \quad (4.22)$$

Calculons la variation de cette énergie dans le temps :

$$\dot{E} = 2\dot{\phi} + m\dot{x}\ddot{x} + m\dot{y}\ddot{y} \quad (4.23)$$

Il est possible d'exprimer la dérivée temporelle du champ en fonction de la force :

$$\dot{\phi} = \frac{\partial \phi}{\partial t} = \frac{\partial \phi}{\partial x} \cdot \frac{\partial x}{\partial t} = \dot{x} \frac{\partial \phi}{\partial x} \quad (4.24)$$

Soit

$$\dot{\phi} = -\dot{x} \cdot F_x \quad (4.25)$$

De manière similaire, on peut aussi affirmer :

$$\dot{\phi} = -\dot{y} \cdot F_y \quad (4.26)$$

En revenant à l'expression de la dynamique virtuelle introduite dans l'équation (3.9) recopiée ici pour commodité :

$$\vec{F} - \lambda \cdot \vec{v} = m \cdot \frac{d\vec{v}}{dt} \quad (4.27)$$

On peut affirmer :

$$\begin{cases} F_x - \lambda \cdot \dot{x} = m \cdot \ddot{x} \\ F_y - \lambda \cdot \dot{y} = m \cdot \ddot{y} \end{cases} \quad (4.28)$$

Et donc, en combinant les équations (4.23), (4.25), (4.26) et (4.28), on obtient :

$$\dot{E} = -\lambda \dot{x}^2 - \lambda \dot{y}^2 \quad (4.29)$$

Pour un coefficient de frottement virtuel positif, on a donc :

$$\dot{E} < 0, \forall (x, y) \neq (0, 0) \quad (4.30)$$

On a donc en tout

$$\begin{cases} E(0,0) = 0 \\ E > 0; \forall (x, y) \neq (0,0) \\ \dot{E} < 0; \forall (x, y) \neq (0,0) \end{cases} \quad (4.31)$$

L'énergie virtuelle  $E$  est donc bien une fonction de Lyapunov. On peut donc conclure que tant que le robot ne s'arrête pas, l'énergie est strictement décroissante. En d'autres termes :

$$E_t < E_{t_0}; \forall t > t_0 \quad (4.32)$$

L'énergie à un instant  $t$  est toujours strictement inférieure à l'énergie initiale du système. Cette propriété est en fait facilement interprétable si on revient à l'analogie mécanique du système : le robot est représenté par un point matériel abandonné à lui-même sur une surface qui présente un frottement visqueux. L'énergie mécanique du système décroît vers 0.

En remplaçant l'expression de l'énergie dans l'équation (4.32) et sachant que le robot part du repos, on obtient :

$$2\phi + \frac{1}{2}m\dot{x}^2 + \frac{1}{2}m\dot{y}^2 < 2\phi_{init} \quad (4.33)$$

Où  $\phi_{init}$  représente le potentiel total du robot lorsqu'il se trouve au point initial. On a donc :

$$\phi < \phi_{init} - \frac{1}{4}m\dot{x}^2 - \frac{1}{4}m\dot{y}^2 \quad (4.34)$$

Et donc, à fortiori

$$\phi < \phi_{init} \quad (4.35)$$

Par conséquent, le potentiel du robot est toujours inférieur au potentiel initial.

On peut donc citer la propriété suivante :

$$\begin{cases} \phi < \phi_m \\ \phi_m = \phi_{init} \end{cases} \quad (4.36)$$

Il est important de noter que cette propriété est vraie quelle que soit la forme des potentiels d'attraction et de répulsion, tant que le potentiel total et que le coefficient de frottement sont positifs. L'existence de la fonction de Lyapunov  $E$  confirme la stabilité de la méthode.

Quelle que soit la forme des potentiels choisis, il est impossible que le robot ne converge pas. Notons que le point de convergence n'est pas nécessairement le but; il peut s'agir d'un minimum local.

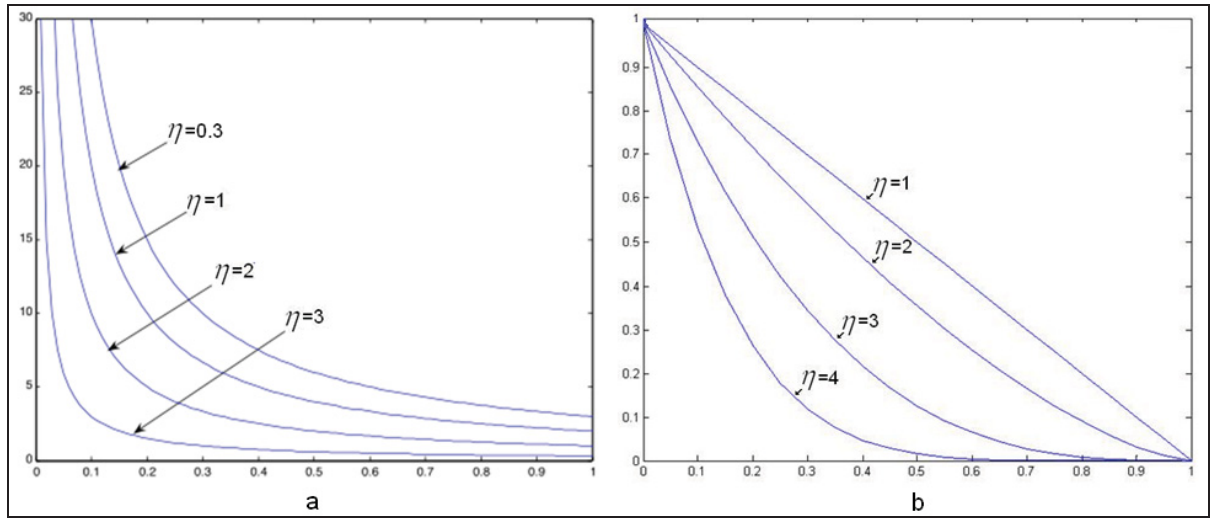
Avec cette propriété du potentiel, on voit bien qu'il n'est pas nécessaire d'imposer un potentiel qui croît indéfiniment lorsque le robot s'approche d'un obstacle. Afin de garantir l'évitement, il suffit d'imposer, à la surface d'un obstacle :

$$\phi_{o,\rho=0} = \phi_m \quad (4.37)$$

Nous introduisons donc la forme finale du potentiel de répulsion des obstacles, qui intègre toutes les propriétés citées précédemment :

$$\phi_o = \phi_m \cdot \left( 1 - e^{-\frac{D_b^2}{R^2}} \right) \cdot \left( \frac{\rho_0 - \rho}{\rho_0} \right)^\eta \quad (4.38)$$

En effet, cette forme du potentiel garantit l'évitement des obstacles puisque, pour  $\rho = 0$ ,  $\phi = \phi_m$ . L'utilisation de l'exposant  $\eta$  permet d'avoir une croissance polynomiale du potentiel de répulsion lorsque le robot s'approche de l'obstacle. En traçant le graphe normalisé du potentiel de répulsion en fonction de la distance  $\rho$ , on obtient les courbes de la Figure 4-3 paramétrées par  $\eta$ . Le choix du paramètre  $\eta$  permet de contrôler la concavité du potentiel de répulsion.



**Figure 4-3 Potentiels de répulsion en fonction de la distance à l'obstacle en grandeurs normalisées. a. potentiel FIRAS - b. forme améliorée du potentiel.**

Finalement, le potentiel total appliqué au centre du robot a les composantes suivantes :

$$\begin{cases} \phi = \phi_o + \phi_b \\ \phi_o = \phi_m \cdot \left(1 - e^{-\frac{D_b^2}{R^2}}\right) \cdot \left(\frac{\rho_0 - \rho}{\rho_0}\right)^\eta \\ \phi_b = \frac{1}{2} \cdot \zeta \cdot [(x - x_b)^2 + (y - y_b)^2] \end{cases} \quad (4.39)$$

En appliquant l'équation (3.3) aux composantes du potentiel, on obtient finalement pour le potentiel d'attraction du but :

$$\begin{cases} F_{b,x} = \zeta(x - x_b) \\ F_{b,y} = \zeta(y - y_b) \end{cases} \quad (4.40)$$

Et pour le potentiel de répulsion des obstacles :

$$\begin{cases} F_{o,x} = -\frac{2\phi_m}{R^2}(x-x_b) \cdot e^{-\frac{D_b^2}{R^2}} \cdot \left(\frac{\rho_0-d_i}{\rho_0}\right)^\eta - \frac{\eta\phi_m}{\rho_0} \cdot \left(1-e^{-\frac{D_b^2}{R^2}}\right) \cdot \left(\frac{\rho_0-d_i}{\rho_0}\right)^{\eta-1} \cos(\theta+\alpha_i) \\ F_{o,y} = -\frac{2\phi_m}{R^2}(y-y_b) \cdot e^{-\frac{D_b^2}{R^2}} \cdot \left(\frac{\rho_0-d_i}{\rho_0}\right)^\eta - \frac{\eta\phi_m}{\rho_0} \cdot \left(1-e^{-\frac{D_b^2}{R^2}}\right) \cdot \left(\frac{\rho_0-d_i}{\rho_0}\right)^{\eta-1} \sin(\theta+\alpha_i) \end{cases} \quad (4.41)$$

Qui peut être approximé, lorsque le robot est loin du but, par :

$$\begin{cases} F_{o,x} = -\frac{\eta\phi_m}{\rho_0} \cdot \left(\frac{\rho_0-d_i}{\rho_0}\right)^{\eta-1} \cos(\theta+\alpha_i) \\ F_{o,y} = -\frac{\eta\phi_m}{\rho_0} \cdot \left(\frac{\rho_0-d_i}{\rho_0}\right)^{\eta-1} \sin(\theta+\alpha_i) \end{cases} \quad (4.42)$$

Notons qu'afin de garantir la continuité de la force, il est nécessaire d'imposer la condition

$$\eta \geq 1 \quad (4.43)$$

#### 4.2.5. Minimisation de l'équilibre instable autour des obstacles

Nous avons vu en 3.2.1 que lorsque le robot a une direction perpendiculaire à la surface de l'obstacle, il a tendance à entrer dans un mouvement oscillatoire pour éviter l'obstacle. Ces oscillations seront mises en évidence au CHAPITRE 6 avec des résultats de simulations et d'expérimentation pratique.

Ce problème est dû au fait que la force qui écarte le robot de sa position d'équilibre instable est très faible par rapport aux forces d'attraction et de répulsion.

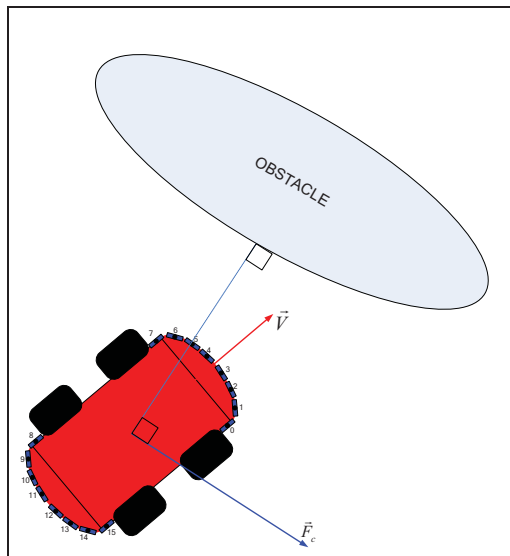
Afin de régler ce problème, il suffit d'ajouter une force tangentielle à la surface de l'obstacle. Cette force devrait faire en sorte que le robot ait tendance à se diriger parallèlement à la surface de l'obstacle plutôt que de s'éloigner de l'obstacle. Dans ces conditions, lorsque le robot s'approche perpendiculairement à l'obstacle, il n'entrera pas dans un mouvement

oscillatoire d'éloignement-avancement, mais suivra une trajectoire plus lisse de contournement tangentiel.

On introduit une force similaire à celle de répulsion radiale, mais dont la direction est tangentielle à la surface de l'obstacle :

$$\begin{cases} F_{c,x} = \text{sgn}(\alpha_i) \cdot \frac{\eta' \cdot \phi_m}{\rho_0} \cdot \left( \frac{\rho_0 - d_i}{\rho_0} \right)^{\eta'-1} \cdot \cos\left(\theta + \alpha_i - \frac{\pi}{2}\right) \\ F_{c,y} = \text{sgn}(\alpha_i) \cdot \frac{\eta' \cdot \phi_m}{\rho_0} \cdot \left( \frac{\rho_0 - d_i}{\rho_0} \right)^{\eta'-1} \cdot \sin\left(\theta + \alpha_i - \frac{\pi}{2}\right) \end{cases} \quad (4.44)$$

Cette expression de la force assure que plus le robot se dirige perpendiculairement à la surface de l'obstacle, plus la force tangentielle est élevée. Cette force est discontinue autour de 0 pour l'angle entre la direction du robot et celui de la force de répulsion radiale, de façon à bien discriminer le sens que doit prendre la trajectoire du robot.



**Figure 4-4 Force tangentielle de contournement d'un obstacle**

Notons que cette force n'est que locale et n'affecte pas la stabilité globale du robot.



### 4.3. Point de vue comportemental

L'utilisation des trois forces  $\vec{F}_b$ ,  $\vec{F}_o$  et  $\vec{F}_c$  peut être interprétée du point de vue comportemental.

En effet, on peut distinguer trois comportements, chacun attribué à une de ces forces :

- un comportement de base qui est de se diriger vers le but qui est représenté par la force  $\vec{F}_b$  ;
- un comportement normal de contournement des obstacles en chemin qui est représenté par la force  $\vec{F}_c$  ;
- un comportement primaire d'évitement d'obstacle qui est représenté par la force  $\vec{F}_o$  .

D'après l'expression de la force de répulsion des obstacles, on voit bien que le robot est repoussé dans la direction radiale par l'obstacle lorsqu'il est dans son rayon d'influence. Cependant, du point de vue comportement, il n'est pas toujours nécessaire que le robot s'éloigne de l'obstacle; en effet, lorsque le robot se déplace parallèlement à la surface de l'obstacle, il n'a pas besoin de l'éviter puisqu'il ne risque pas de rentrer en collision avec lui. L'expression obtenue pour la force de répulsion ne distingue pas cette nuance et repousse le robot dans tous les cas.

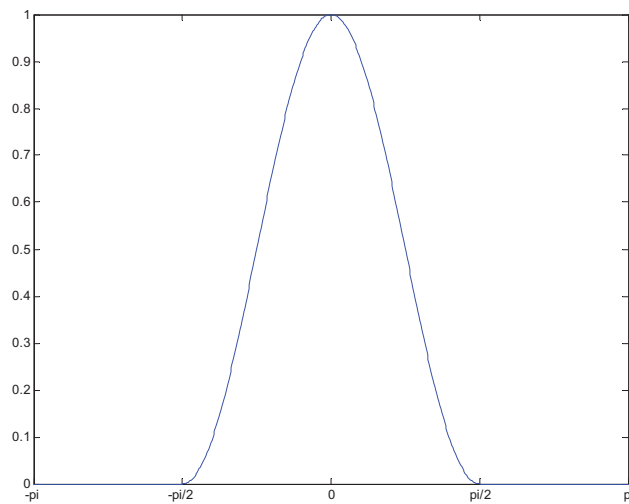
Il est donc intéressant de moduler la force de répulsion  $\vec{F}_o$  en fonction de l'angle  $\alpha$  qui représente la direction du robot par rapport à la surface de l'obstacle rencontré. Cette force devrait être continue, maximale lorsque le robot se déplace perpendiculairement à la surface de l'obstacle et nulle lorsque le robot se dirige parallèlement à cette dernière; elle devrait aussi être nulle lorsque le robot s'éloigne de l'obstacle.

Il suffit pour cela de multiplier la force de répulsion indépendante de la direction de déplacement du robot par une fonction de modulation. On désigne par  $\mu_o$  la fonction de

modulation de la force radiale de répulsion. Plusieurs fonctions répondant au critère imposé peuvent être choisies. La fonction que l'on introduit a pour expression :

$$\mu_o(\alpha) = \begin{cases} \cos^2 \alpha; \alpha \in \left[ -\frac{\pi}{2} + 2k\pi; \frac{\pi}{2} + 2k\pi \right] \\ 0; \alpha \notin \left[ -\frac{\pi}{2} + 2k\pi; \frac{\pi}{2} + 2k\pi \right] \end{cases} \quad (4.45)$$

Cette expression permet d'avoir une fonction de modulation continue; elle permet aussi d'en avoir la dérivée continue, de façon à éviter les « secousses ». La Figure 4-5 représente le graphe de la fonction  $\mu_o$ .



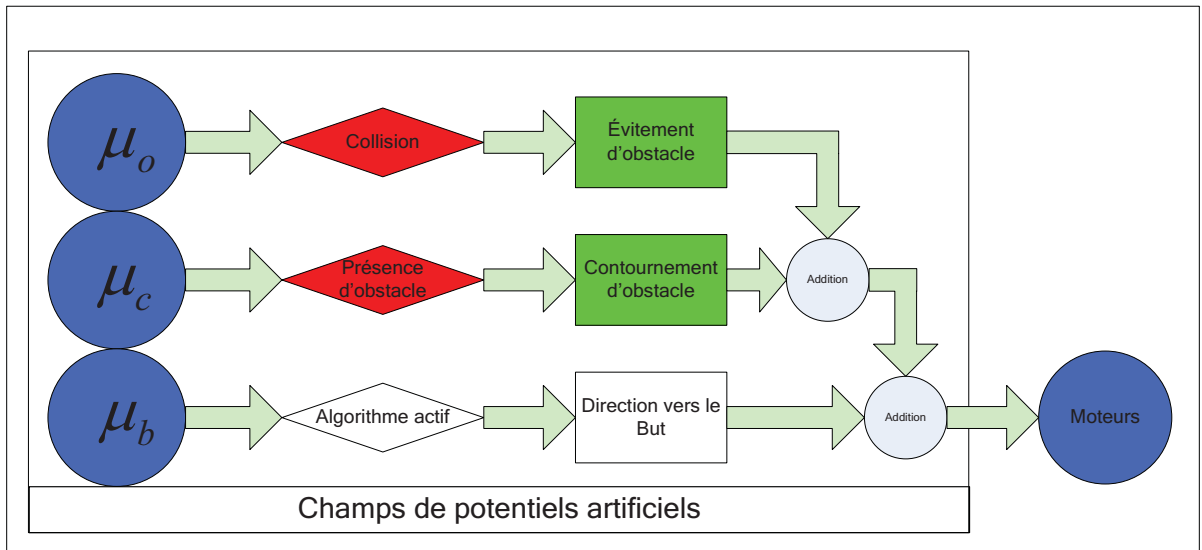
**Figure 4-5 Allure de la fonction de modulation de la force de répulsion**

En utilisant les fonctions de modulation, il est possible d'exprimer la méthode de champs de potentiels artificiels décrite dans ce chapitre d'un point de vue comportemental. Puisqu'en tout moment il est possible pour le robot de subir n'importe laquelle des forces décrites, chacune avec une intensité donnée, les fonctions de modulation représentent donc le niveau d'activation de chaque comportement. En désignant par  $\mu_c$  la fonction de modulation de la force de contournement et par  $\mu_b$  celle de la force d'attraction du but, on peut dire :

$$\begin{aligned}\mu_b &= 1 \\ \mu_c &= H\left(\frac{d}{\rho_0}\right) \\ \mu_o(\alpha) &= \begin{cases} \cos^2 \alpha; \alpha \in \left[-\frac{\pi}{2} + 2k\pi; \frac{\pi}{2} + 2k\pi\right] \\ 0; \alpha \notin \left[-\frac{\pi}{2} + 2k\pi; \frac{\pi}{2} + 2k\pi\right] \end{cases}\end{aligned}\quad (4.46)$$

Où  $H$  désigne la fonction échelon.

Finalement, on peut donc parler de comportements continus : à chaque instant, chacun des comportements possibles est exécuté à un certain pourcentage qui n'est autre que la fonction de modulation de la force associée. Schématiquement, on peut représenter le diagramme du comportement vu à la Figure 3-2 dans celui de la Figure 4-6.



**Figure 4-6 Diagramme de comportements continus**

#### **4.4. Conclusion**

En tout, on a élaboré dans ce chapitre la méthode des champs de potentiels artificiels de façon à faire fonctionner le robot dans un environnement inconnu. Les limitations des méthodes classiques de champs de potentiels artificiels ont été levées. Il reste la limitation concernant les minima locaux qui n'a pas été traitée dans ce rapport à cause du manque de temps qui a limité l'expérimentation; cette méthode a déjà été traitée avec succès dans la littérature - Park et Lee (2003).

La méthode a aussi été interprétée d'un point de vue comportemental, et le concept de comportements continus a été introduit.

Dans le prochain chapitre nous traitons la problématique du choix des paramètres introduite à la section 3.2.1.

## CHAPITRE 5

### CHOIX OBJECTIF DES PARAMÈTRES

#### 5.1. Introduction

Dans ce chapitre nous abordons le problème du choix des paramètres de l'algorithme. En effet, lors de l'élaboration des équations utilisées dans l'algorithme de navigation, les paramètres suivants ont été introduits :

$$\eta; \eta'; \zeta; \lambda; m \quad (5.1)$$

Dans la littérature, les méthodes utilisant les champs de potentiels choisit la valeur des paramètres par essai et erreur, de façon à ce que la trajectoire suivie soit « logique ».

Dans ce chapitre, nous tentons de limiter autant que possible les approches par essai et erreur en introduisant des critères objectifs de choix des paramètres.

#### 5.2. Calcul de $\zeta$ et de $\lambda$

En revenant aux équations du système, on a :

$$\begin{cases} F_x - \lambda V_x = m \dot{V}_x \\ F_y - \lambda V_y = m \dot{V}_y \end{cases} \quad (5.2)$$

Vu que l'on ne connaît rien sur l'environnement à priori, on commence par supposer que le robot se trouve suffisamment loin des obstacles pour en négliger le champ. Cette supposition sera validée plus loin avec le choix de  $\eta$ .

Dans ces conditions, on peut exprimer la force virtuelle subie par le robot :

$$\begin{cases} F_x = -\zeta(x - x_b) \\ F_y = -\zeta(y - y_b) \end{cases} \quad (5.3)$$

Ce qui donne un système linéaire du second ordre :

$$\begin{cases} \ddot{x} + \frac{\lambda}{m} \dot{x} + \frac{\zeta}{m} x = \frac{\zeta}{m} x_b \\ \ddot{y} + \frac{\lambda}{m} \dot{y} + \frac{\zeta}{m} y = \frac{\zeta}{m} y_b \end{cases} \quad (5.4)$$

En exprimant le système sous forme de fonction de transfert, pour des conditions initiales nulles (puisque le robot part de l'origine) on obtient :

$$\begin{cases} X(s) = \frac{\frac{\zeta}{m}}{s^2 + \frac{\lambda}{m}s + \frac{\zeta}{m}} \cdot X_b(s) \\ Y(s) = \frac{\frac{\zeta}{m}}{s^2 + \frac{\lambda}{m}s + \frac{\zeta}{m}} \cdot Y_b(s) \end{cases} \quad (5.5)$$

En identifiant cette écriture à la forme canonique d'un système du second ordre,

$$H(s) = \frac{K\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (5.6)$$

On obtient :

$$\begin{cases} \zeta = m\omega_n^2 \\ \lambda = 2m\xi\omega_n \end{cases} \quad (5.7)$$

Pour calculer  $\zeta$  et  $\lambda$  il faut donc trouver une manière de fixer le facteur d'amortissement ainsi que la pulsation naturelle du système. Pour cela nous utilisons deux contraintes.

La première contrainte est relative aux oscillations. En regardant le système en tant que système du second ordre de sorties  $x$  et  $y$ , on peut dire que l'on cherche à avoir le dépassement le plus faible possible : on veut que le robot arrive directement au but, sans le dépasser et y retourner. Contrairement aux systèmes classiques du second ordre, ce qui nous intéresse ici n'est pas le pourcentage de dépassement, mais bien le dépassement absolu. En effet, quelle que soient la position du but et la position initiale du robot, le robot devrait toujours s'arrêter net au but.

On commence donc par se fixer un dépassement absolu tolérable, noté  $D_a$ . Le pourcentage de dépassement est donc :

$$\begin{cases} D_{\%,x} = \frac{D_a}{x_b - x_0} \\ D_{\%,y} = \frac{D_a}{y_b - y_0} \end{cases} \quad (5.8)$$

Il suffit de choisir ici le pourcentage de dépassement le plus petit, c'est-à-dire aussi :

$$D_{\%} = \frac{D_a}{\max(x_b - x_0, y_b - y_0)} \quad (5.9)$$

La seconde contrainte à considérer est celle du temps de parcours. On considère idéalement que le robot parcourt à sa vitesse maximale la distance entre la position initiale et le but. Le temps de montée du système, noté  $T_p$ , est donc donné par la formule :

$$T_p = \frac{\sqrt{(x_b - x_0)^2 + (y_b - y_0)^2}}{V_{\max}} \quad (5.10)$$

Une fois le pourcentage de dépassement et le temps de montée calculés, on peut déduire, en utilisant les formules des systèmes du second ordre – Nise (2003), les expressions du facteur d'amortissement et de la pulsation naturelle du système :

$$\left\{ \begin{array}{l} \xi = \frac{\ln \frac{\max(x_b - x_0, y_b - y_0)}{D_a}}{\sqrt{\pi^2 + \ln^2 \frac{\max(x_b - x_0, y_b - y_0)}{D_a}}} \\ \omega_n = \sqrt{\frac{\pi^2 + \ln^2 \frac{\max(x_b - x_0, y_b - y_0)}{D_a}}{(x_b - x_0)^2 + (y_b - y_0)^2}} \cdot V_{\max} \end{array} \right. \quad (5.11)$$

Ce qui donne finalement les valeurs optimales de  $\lambda$  et de  $\zeta$  :

$$\left\{ \begin{array}{l} \lambda = 2m \frac{\ln \frac{\max(x_b - x_0, y_b - y_0)}{D_a}}{\sqrt{(x_b - x_0)^2 + (y_b - y_0)^2}} \cdot V_{\max} \\ \zeta = m \frac{\pi^2 + \ln^2 \frac{\max(x_b - x_0, y_b - y_0)}{D_a}}{(x_b - x_0)^2 + (y_b - y_0)^2} \cdot V_{\max} \end{array} \right. \quad (5.12)$$

### 5.3. Calcul de $\eta$ et $\eta'$

Ici, comme nous n'avons pas d'information sur l'environnement, nous effectuons le raisonnement suivant :

On cherche à garder le comportement du robot le plus proche possible de celui prévu sans obstacles. On doit donc choisir le paramètre  $\eta$  le plus grand possible, de façon à ce que le



potentiel de répulsion soit le plus faible possible : Plus  $\eta$  est grand, moins l'influence d'un obstacle est importante. Cela veut dire aussi que lorsque les obstacles sont proches les uns des autres, le robot a moins de risque d'être pris dans un minimum local lorsque  $\eta$  est grand.

Par contre, lorsque le paramètre  $\eta$  est trop grand, la pente du potentiel aux alentours de l'obstacle devient élevée, ce qui fait que la force de répulsion est élevée. Le robot acquiert alors une accélération très importante. Il s'agit donc de choisir la plus grande valeur possible de  $\eta$ , tout en gardant l'accélération du robot à la répulsion inférieure à une accélération maximale  $a_m$ .

En revenant au potentiel d'un obstacle,

$$\phi_o = \phi_m \cdot \left( 1 - e^{-\frac{D_b^2}{R^2}} \right) \cdot \left( \frac{\rho_0 - \rho}{\rho_0} \right)^\eta \quad (5.13)$$

Puisque l'on a

$$0 < d_i < \rho_0 \quad (5.14)$$

Et comme on considère avant d'atteindre le but que :

$$1 - e^{-\frac{D_b^2}{R^2}} \approx 1 \quad (5.15)$$

On peut affirmer que le potentiel  $\phi_o$  est toujours inférieur au potentiel  $\phi_m$ , l'égalité ayant lieu au contact avec l'obstacle.

Afin de choisir le paramètre  $\eta$ , on se place dans le pire des cas de la force de répulsion, c'est-à-dire lorsque le robot est en contact avec un obstacle et qu'il se dirige droit vers lui. Dans ces conditions, la force de contournement est nulle. En réécrivant la force radiale :

$$\left\{ \begin{array}{l} F_{o,x} = -\frac{2\phi_m}{R^2}(x-x_b) \cdot e^{-\frac{D_b^2}{R^2}} \cdot \left(\frac{\rho_0-d_i}{\rho_0}\right)^\eta \\ \quad - \frac{\eta\phi_m}{\rho_0} \cdot \left(1 - e^{-\frac{D_b^2}{R^2}}\right) \cdot \left(\frac{\rho_0-d_i}{\rho_0}\right)^{\eta-1} \cos(\theta + \alpha_i) \\ F_{o,y} = -\frac{2\phi_m}{R^2}(y-y_b) \cdot e^{-\frac{D_b^2}{R^2}} \cdot \left(\frac{\rho_0-d_i}{\rho_0}\right)^\eta \\ \quad - \frac{\eta\phi_m}{\rho_0} \cdot \left(1 - e^{-\frac{D_b^2}{R^2}}\right) \cdot \left(\frac{\rho_0-d_i}{\rho_0}\right)^{\eta-1} \sin(\theta + \alpha_i) \end{array} \right. \quad (5.16)$$

Et en considérant que le robot est au contact d'un obstacle et qu'il est suffisamment loin du but, c'est-à-dire :

$$\left\{ \begin{array}{l} d_i = 0 \\ \left(1 - e^{-\frac{D_b^2}{R^2}}\right) \approx 1 \end{array} \right. \quad (5.17)$$

On obtient :

$$\left\{ \begin{array}{l} F_{o,x} = -\frac{\eta\phi_m}{\rho_0} \cdot \cos(\theta + \alpha_i) \\ F_{o,y} = -\frac{\eta\phi_m}{\rho_0} \cdot \sin(\theta + \alpha_i) \end{array} \right. \quad (5.18)$$

La norme de la force de répulsion est :

$$F_o = \sqrt{F_{x,o}^2 + F_{y,o}^2} \quad (5.19)$$

Donc

$$F_o = \frac{\eta \phi_m}{\rho_0} \quad (5.20)$$

En revenant à l'équation du mouvement

$$F - \lambda V = m\dot{V} \quad (5.21)$$

Et en considérant qu'au contact de l'obstacle, le robot rebrousse chemin, c'est-à-dire

$$V = 0 \quad (5.22)$$

On obtient la relation

$$F = m\dot{V} = ma \quad (5.23)$$

Si on ne considère que la force de répulsion, en remplaçant la force par son expression, on obtient :

$$\frac{\eta \phi_m}{\rho_0} = ma \quad (5.24)$$

Si on impose une accélération maximale  $a_m$ , on peut alors écrire à la limite :

$$\eta = \frac{ma_m \rho_0}{\phi_m} \quad (5.25)$$

Notons que puisque le paramètre  $\eta$  doit être supérieur à 1, on a donc maintenant une condition sur le rayon maximal  $\rho_0$  :

$$a_m \rho_0 > \frac{\phi_m}{m} \quad (5.26)$$

De la même façon que pour le paramètre  $\eta$ , on peut établir que lorsque le robot se trouve à la limite du contact avec l'obstacle et qu'il est dirigé parallèlement à sa surface, le module de la force de contournement  $\vec{F}_c$  devient :

$$F_c = \frac{\eta' \phi_m}{\rho_0} \quad (5.27)$$

En imposant la même contrainte de l'accélération maximale imposée pour le calcul de  $\eta$ , on arrive à la conclusion :

$$\eta' = \eta = \frac{ma_m \rho_0}{\phi_m} \quad (5.28)$$

En revenant à l'expression du potentiel initial  $\phi_m$  :

$$\phi_m = \frac{1}{2} \cdot \zeta \cdot \left[ (x_0 - x_b)^2 + (y_0 - y_b)^2 \right] \quad (5.29)$$

Et en notant que le paramètre  $\zeta$  est proportionnel à  $m$  selon l'équation (5.12), on peut remplacer les termes par leur valeur dans l'équation (5.28), ce qui donne :

$$\eta' = \eta = \frac{a_m \rho_0 \left( (x_b - x_0)^2 + (y_b - y_0)^2 \right)}{V_{\max} \left( \pi^2 + \ln^2 \frac{\max(x_b - x_0, y_b - y_0)}{D_a} \right)} \quad (5.30)$$

On peut en tirer que les paramètres  $\eta$  et  $\eta'$  ne dépendent pas de  $m$ .

#### 5.4. Calcul de $m$

Revenons à l'équation qui donne la référence de vitesse (5.21). En remplaçant les forces et les paramètres par leur expression, et en approximant la force d'évitement  $\vec{F}_o$  lorsque le robot se trouve loin du but, on obtient pour l'axe des abscisses :

$$\begin{aligned} & -m \cdot \frac{\pi^2 + \ln^2 \frac{\max(x_b - x_0, y_b - y_0)}{D_a}}{(x_b - x_0)^2 + (y_b - y_0)^2} \cdot V_{\max} \cdot (x - x_0) \\ & - \frac{1}{2} \cdot m \cdot \frac{\pi^2 + \ln^2 \frac{\max(x_b - x_0, y_b - y_0)}{D_a}}{(x_b - x_0)^2 + (y_b - y_0)^2} \cdot \left[ (x_0 - x_b)^2 + (y_0 - y_b)^2 \right] \cdot \frac{\eta}{\rho_0} \cdot \left( \frac{\rho_0 - d_i}{\rho_0} \right)^{\eta-1} \cdot \cos(\theta + \alpha_i) \\ & + \operatorname{sgn}(\alpha_i) \cdot m \cdot \frac{\pi^2 + \ln^2 \frac{\max(x_b - x_0, y_b - y_0)}{D_a}}{(x_b - x_0)^2 + (y_b - y_0)^2} \cdot \left[ (x_0 - x_b)^2 + (y_0 - y_b)^2 \right] \cdot \frac{\eta'}{\rho_0} \cdot \left( \frac{\rho_0 - d_i}{\rho_0} \right)^{\eta'-1} \cdot \cos\left(\theta + \alpha_i - \frac{\pi}{2}\right) \\ & - 2 \cdot m \cdot \frac{\ln \frac{\max(x_b - x_0, y_b - y_0)}{D_a}}{\sqrt{(x_b - x_0)^2 + (y_b - y_0)^2}} \cdot V_{\max} \cdot \dot{x} = m\ddot{x} \end{aligned} \quad (5.31)$$

Donc, en simplifiant par le paramètre  $m$  :

$$\begin{aligned}
\ddot{x} = & -\frac{\pi^2 + \ln^2 \frac{\max(x_b - x_0, y_b - y_0)}{D_a}}{(x_b - x_0)^2 + (y_b - y_0)^2} \cdot V_{\max} \cdot (x - x_0) \\
& - \frac{1}{2} \cdot \frac{\pi^2 + \ln^2 \frac{\max(x_b - x_0, y_b - y_0)}{D_a}}{(x_b - x_0)^2 + (y_b - y_0)^2} \cdot [(x_0 - x_b)^2 + (y_0 - y_b)^2] \cdot \frac{\eta}{\rho_0} \cdot \left( \frac{\rho_0 - d_i}{\rho_0} \right)^{\eta-1} \cdot \cos(\theta + \alpha_i) \\
& + \operatorname{sgn}(\alpha_i) \cdot \frac{\pi^2 + \ln^2 \frac{\max(x_b - x_0, y_b - y_0)}{D_a}}{(x_b - x_0)^2 + (y_b - y_0)^2} \cdot [(x_0 - x_b)^2 + (y_0 - y_b)^2] \cdot \frac{\eta'}{\rho_0} \cdot \left( \frac{\rho_0 - d_i}{\rho_0} \right)^{\eta'-1} \cdot \cos\left(\theta + \alpha_i - \frac{\pi}{2}\right) \\
& - 2 \cdot \frac{\ln \frac{\max(x_b - x_0, y_b - y_0)}{D_a}}{\sqrt{(x_b - x_0)^2 + (y_b - y_0)^2}} \cdot V_{\max} \cdot \dot{x}
\end{aligned} \tag{5.32}$$

L'équation obtenue ne dépend pas de  $m$ . La même conclusion peut être tirée avec l'équation du mouvement selon l'axe des ordonnées. Donc la valeur de  $m$  n'a aucune influence sur le mouvement du robot.

Finalement, il suffit de fixer  $m = 1$ , ce qui revient à ne pas utiliser ce paramètre dans les équations.

Finalement, on tire le choix objectif des trois paramètres qui nous intéressent :

$$\left\{ \begin{array}{l} \lambda = 2 \frac{\ln \frac{\max(x_b - x_0, y_b - y_0)}{D_a}}{\sqrt{(x_b - x_0)^2 + (y_b - y_0)^2}} \cdot V_{\max} \\ \zeta = \frac{\pi^2 + \ln^2 \frac{\max(x_b - x_0, y_b - y_0)}{D_a}}{(x_b - x_0)^2 + (y_b - y_0)^2} \cdot V_{\max} \\ \eta = \frac{a_m \rho_0}{\phi_m} \\ \eta' = \frac{a_m \rho_0}{\phi_m} \end{array} \right. \tag{5.33}$$

## CHAPITRE 6

### RÉSULTATS

#### 6.1. Introduction

Dans ce chapitre on met à l'épreuve la théorie développée dans les chapitres précédents. On effectue deux séries d'essais :

- simulations sur Matlab;
- essais sur un robot réel.

Dans le cas des simulations, on commence par comparer les performances de la forme modifiée du potentiel de répulsion par rapport à la forme classique. On évalue ensuite l'amélioration due à l'utilisation de la force de contournement. On détaille enfin l'influence de chaque paramètre sur la performance pour montrer l'amélioration des performances quand les paramètres sont calculés objectivement.

Dans le cas des essais sur le robot réel, on évalue l'amélioration qu'apporte la force de contournement puis on confirme l'amélioration apportée par le calcul objectif des paramètres.

#### 6.2. Critères de comparaison

Afin de pouvoir mesurer objectivement la performance de l'algorithme dans chaque condition, il est important de spécifier des critères objectifs qui permettent d'en mesurer facilement l'efficacité. On introduit les trois paramètres suivants :

- longueur totale du parcours  $L_t$  ;
- durée du parcours  $D_p$  ;
- coefficient d'oscillations  $C_o$  .

### 6.2.1. Longueur totale du parcours

Lorsqu'on cherche à concevoir un algorithme de navigation performant, un critère d'optimisation est logiquement la distance que parcourt le robot entre le point de départ et le point d'arrivée. En d'autres termes, la longueur de la trajectoire est une mesure de l'efficacité d'un algorithme de navigation. Dans notre cas, plus la trajectoire est courte, plus l'algorithme est performant. Cette distance  $L_t$  est approximée numériquement par :

$$L_t = \sum_{k=0}^N v_k \cdot \tau \quad (6.1)$$

Où  $N$  est le nombre de pas de calcul entre l'instant de départ du robot et l'instant de son arrivée au but.  $v_k$  représente le module de la vitesse à l'instant  $k$ .  $\tau$  représente la durée du pas de calcul.

### 6.2.2. Durée totale du parcours

Il est possible pour le robot de suivre la même trajectoire avec des vitesses différentes. Afin de bien rendre compte de la vitesse, on introduit le temps total de parcours qui, théoriquement, devrait être le plus proche possible du temps idéal considéré sans obstacles. Le second paramètre d'évaluation est donc :

$$D_p = N \cdot \tau \quad (6.2)$$

### 6.2.3. Coefficient d'oscillations

Vu qu'une des caractéristiques principales d'une trajectoire générée par la méthode des champs de potentiels est l'oscillation, il est intéressant de fixer un critère qui illustre le niveau d'oscillations de la trajectoire. En effet, le robot oscille lorsqu'il change de direction. En d'autres termes, c'est la présence de la vitesse de rotation  $\omega$  qui indique que le robot



oscille. La mesure de la moyenne quadratique des vitesses instantanées de rotation  $C_o$  constitue donc un critère intéressant pour la mesure des oscillations de la trajectoire :

$$C_o = \frac{1}{N} \sqrt{\sum_{k=0}^N \omega_k^2} \quad (6.3)$$

Il faut noter que le facteur  $C_o$  ne reflète pas exclusivement les oscillations; il reflète aussi les changements de direction du robot dus simplement au contournement des obstacles. Vu que le robot évite les obstacles dans tous les cas, on peut considérer que cette influence est de même ordre pour tous les essais dans le même environnement. Notons aussi qu'une bonne trajectoire implique moins de changements de direction, ce qui justifie quand même l'utilisation de ce coefficient.

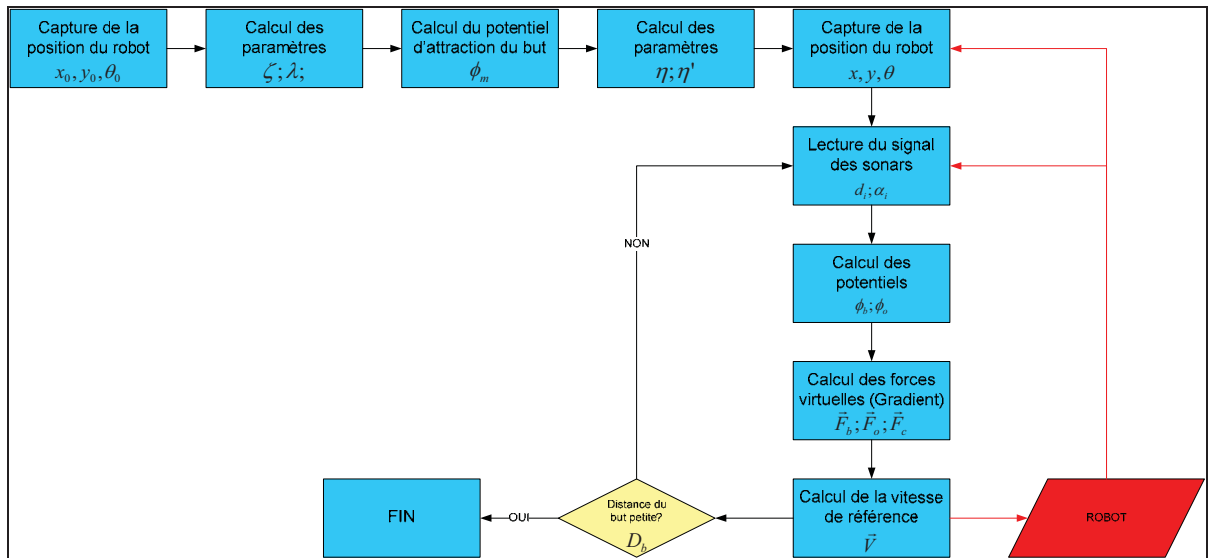
### 6.3. Simulations

Afin d'appliquer la théorie développée, il est important de spécifier un cadre d'expérimentation et de réécrire les équations de façon à pouvoir les programmer numériquement.

Commençons par résumer l'algorithme de navigation théorique. À chaque pas de calcul, les étapes suivantes seraient exécutées :

1. Calcul des potentiels  $\phi_b$  et  $\phi_o$  ;
2. Calcul des forces par dérivation spatiale des potentiels respectifs;
3. Application de l'équation du mouvement pour calculer la vitesse de référence.

Schématiquement, l'algorithme est détaillé dans le diagramme de la Figure 6-1.



**Figure 6-1 Algorithme de navigation**

La simulation sous Matlab a les caractéristiques suivantes :

- le robot est considéré comme un point matériel;
- la distance du robot aux obstacles est calculée précisément; ceci est équivalent à dire que le robot est entouré par une infinité de sonars qui captent dans toutes les directions;
- l'environnement du robot est divisé en parcelles élémentaires. Chaque parcelle est soit libre soit contient un obstacle. La carte désirée est convertie à partir d'un dessin monochrome en une matrice de 0 et de 1 (0 pour une parcelle libre et 1 pour une parcelle occupée par un obstacle);
- l'algorithme s'arrête une fois le robot devient suffisamment proche du but. La distance de tolérance est configurable mais n'a pas d'impact sur les performances de l'algorithme. Elle sera fixée à 30cm dans tous les essais.

### 6.3.1. Forces de répulsions et de contournement

Dans cette partie, on compare entre elles les performances des différentes formes de forces de répulsion. La première force illustrée est la forme *FIRAS* (3.6). On illustre ensuite la

nouvelle forme de potentiel introduite dans ce projet (4.38), sans introduire la force de contournement. On illustre finalement la présence de la force de contournement avec la nouvelle forme de potentiel.

Vu que les paramètres  $\lambda$  et  $\zeta$  agissent sur le potentiel d'attraction du but et n'influent pas sur le potentiel de répulsion des obstacles, une valeur commode leur sera attribuée (identique pour les trois cas) :

$$\begin{cases} \lambda = 0.38 \\ \zeta = 0.07 \end{cases} \quad (6.4)$$

Ces valeurs sont obtenues en utilisant les équations du système (5.33) en choisissant :

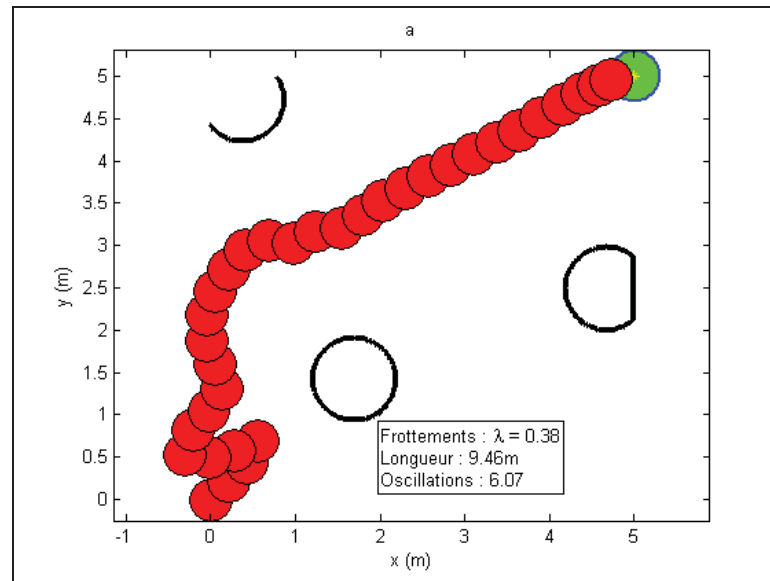
$$\begin{aligned} V_m &= 0.3m/s \\ D_a &= 10cm \\ x_b &= 5m \\ y_b &= 5m \end{aligned} \quad (6.5)$$

Puisque la signification du paramètre  $\eta$  n'est pas la même pour les deux formes du potentiel de répulsion, plusieurs essais seront effectués pour couvrir plus ou moins une plage significative de valeurs de ce paramètre.

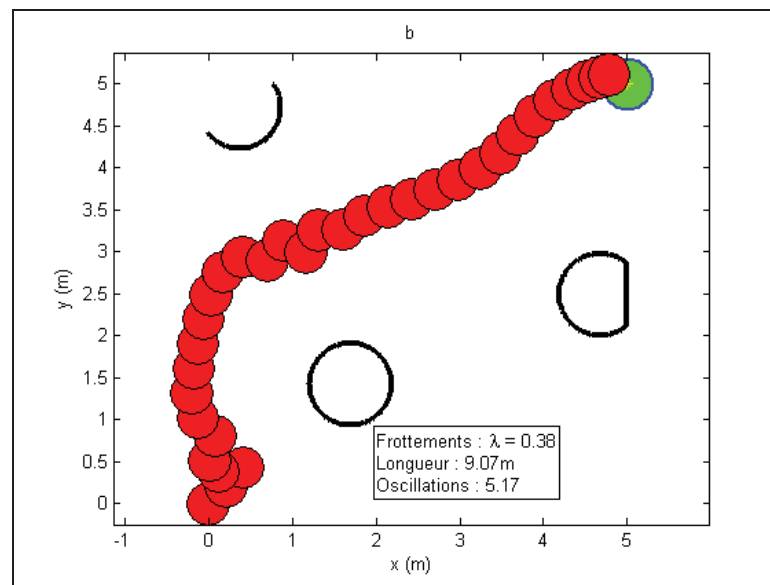
Pour chaque essai effectué, on montre une figure de la trajectoire du robot. Ce graphique représente le robot par un disque rouge, la position du but par un point jaune entouré par un disque vert pour représenter la zone de tolérance et les obstacles par des contours noirs.

#### 6.3.1.1. Forme classique de la force de contournement

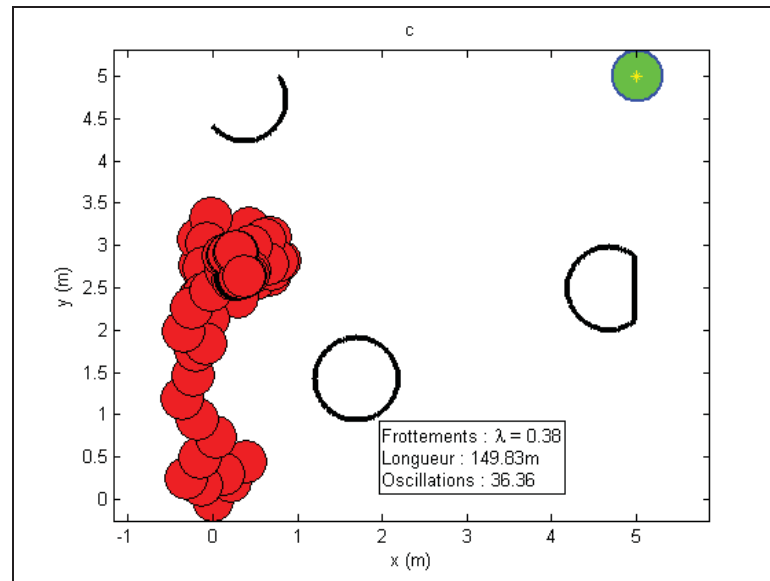
On commence par simuler l'algorithme en utilisant la forme FIRAS du potentiel de répulsion des obstacles. On effectue les essais pour trois valeurs différentes du paramètre  $\eta$ .



**Figure 6-2**    **Forme FIRAS du potentiel de répulsion -  $\eta = 1$**



**Figure 6-3**    **Forme FIRAS du potentiel de répulsion -  $\eta = 2$**



**Figure 6-4** Forme FIRAS du potentiel de répulsion -  $\eta = 4$

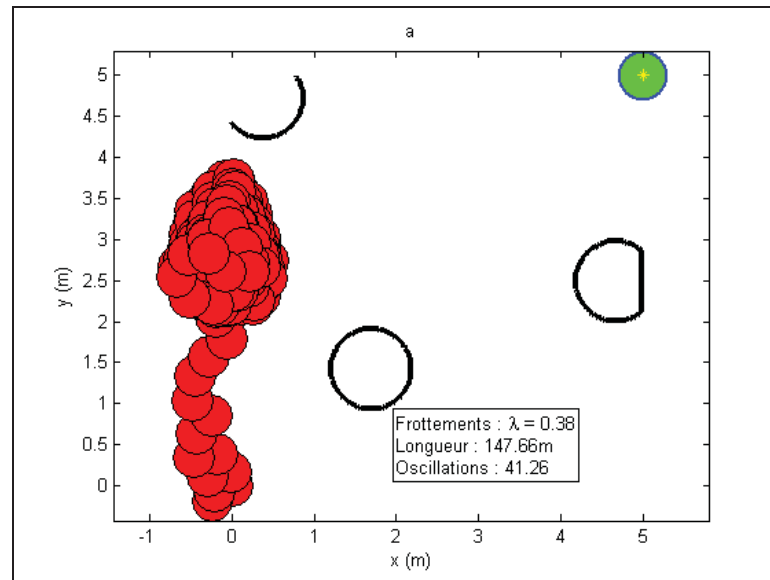
Le graphe de la Figure 6-2 montre comment l'utilisation de la fonction classique FIRAS permet bien de mener le robot à destination. En examinant la Figure 6-3, on voit que lorsque la valeur du paramètre  $\eta$  augmente, les oscillations de la trajectoire augmentent. La Figure 6-4 montre qu'une valeur trop grande de ce paramètre peut piéger le robot dans un minimum local

Le graphe de la Figure 6-3 montre que lorsque l'algorithme utilise la fonction de répulsion FIRAS, le robot a tendance à avoir des oscillations exagérées devant les obstacles.

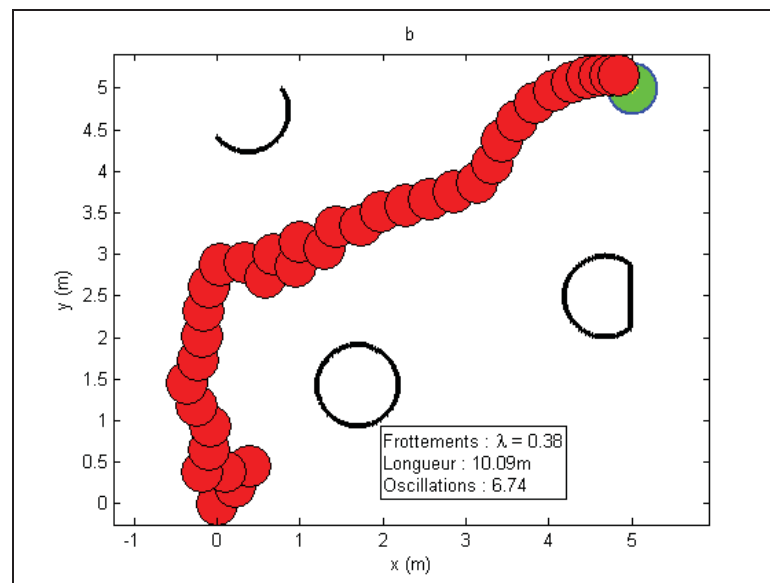
Lorsque le coefficient  $\eta$  devient trop grand, la Figure 6-4 montre que le potentiel entre obstacles voisins augmente de façon à piéger le robot dans un minimum local. Si aucun comportement prenant en charge les minima locaux n'est implémenté, le robot oscillera continuellement autour d'un point qui n'est pas le but.

### 6.3.1.2. Forme modifiée du potentiel de répulsion

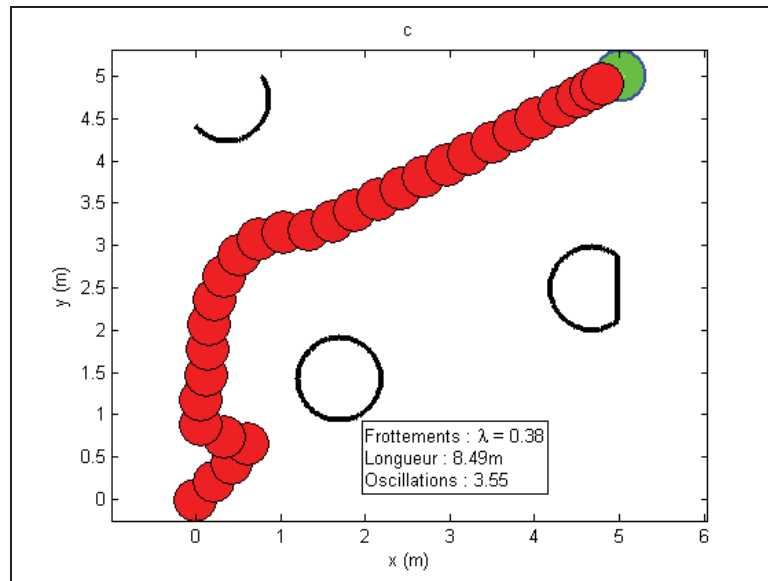
Dans cette partie, on effectue des essais en utilisant la nouvelle forme du potentiel de répulsion. On effectue trois essais, chacun pour une valeur différente du paramètre  $\eta$ .



**Figure 6-5** Nouvelle forme du potentiel de répulsion -  $\eta = 1$



**Figure 6-6** Nouvelle forme du potentiel de répulsion -  $\eta = 2$



**Figure 6-7** Nouvelle forme du potentiel de répulsion -  $\eta = 4$

L'utilisation de la nouvelle forme du potentiel améliore le comportement du robot au voisinage d'un obstacle. En effet, la limitation du potentiel maximal à  $\phi_m$  permet de garder la force de répulsion dans une plage de variations plus raisonnable. La Figure 6-6 et la Figure 6-7 montrent des variations du potentiel de répulsion plus douces que celles de la Figure 6-2 ou de la Figure 6-3.

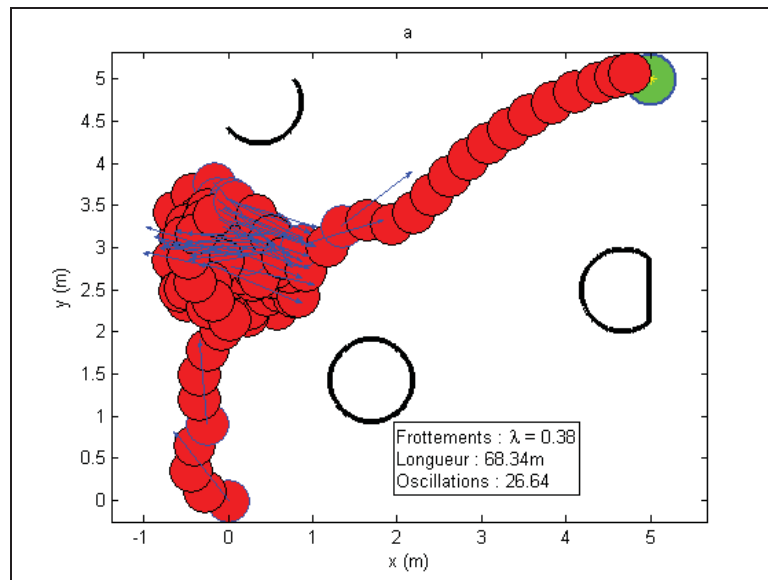
Dans ce cas, la signification du paramètre  $\eta$  est différente et ce sont les petites valeurs de ce paramètre qui élargissent l'influence des obstacles au lieu des grandes valeurs dans le cas précédent. Il n'est donc pas utile de comparer les deux méthodes pour les mêmes valeurs du paramètre  $\eta$ . Ceci ne pose cependant pas de problème pour l'interprétation des résultats.

Le comportement du robot dans ce cas semble cependant avoir la même sensibilité aux minima locaux. En effet, la nouvelle forme du potentiel de répulsion ne permet pas d'éviter la formation de « murs virtuels » autour des obstacles proches l'un de l'autre comme le montre la Figure 6-5.

### 6.3.1.3. Utilisation de la force de contournement

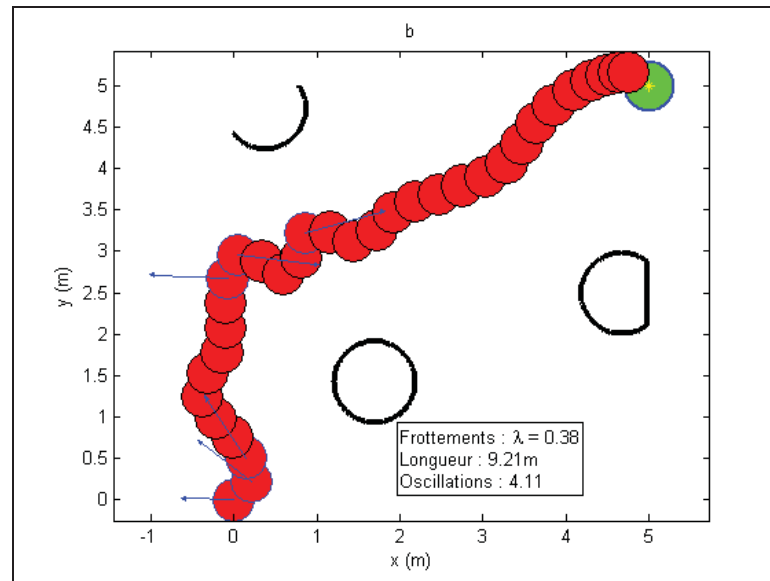
En combinant la nouvelle forme du potentiel de répulsion et l'utilisation de la force de contournement, on réussit à améliorer grandement la forme de la trajectoire.

Dans les graphiques suivants, la direction de la force de contournement est indiquée par des flèches bleues.

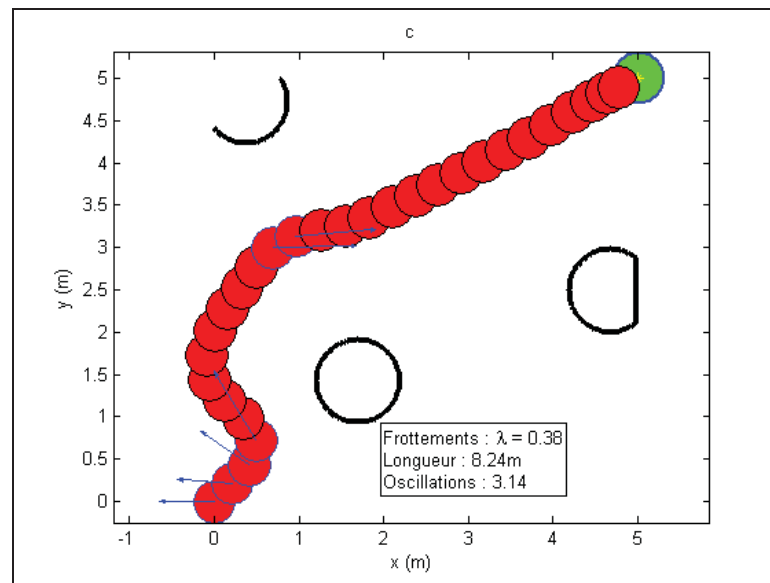


**Figure 6-8** Nouvelle forme du potentiel de répulsion avec force de contournement -  $\eta = 1$





**Figure 6-9** Nouvelle forme du potentiel de répulsion avec force de contournement -  $\eta = 2$



**Figure 6-10** Nouvelle forme du potentiel de répulsion avec force de contournement -  $\eta = 4$

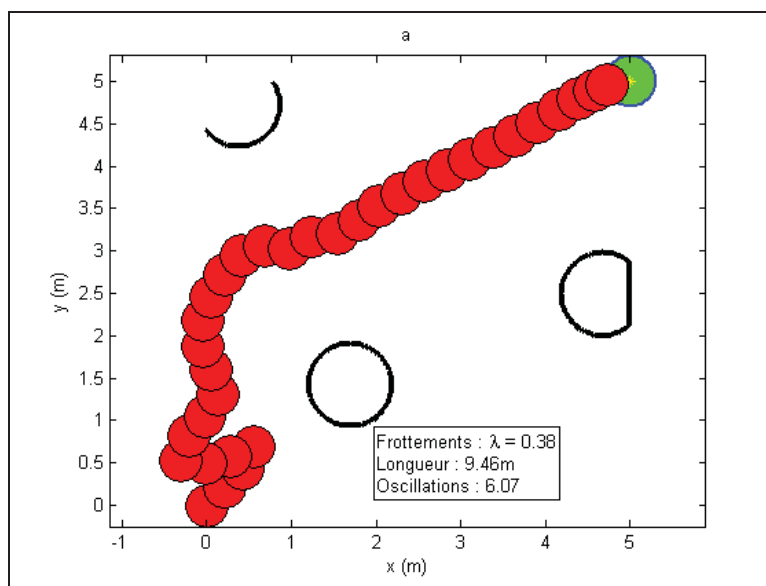
La Figure 6-9 et la Figure 6-10 montrent clairement l'avantage apporté par l'ajout de la force de contournement. En effet, lorsque le robot se dirige vers un obstacle, une force tangentielle

à ce dernier est appliquée au robot, ce qui adoucit la réponse du robot. Il y a moins d'oscillations inutiles.

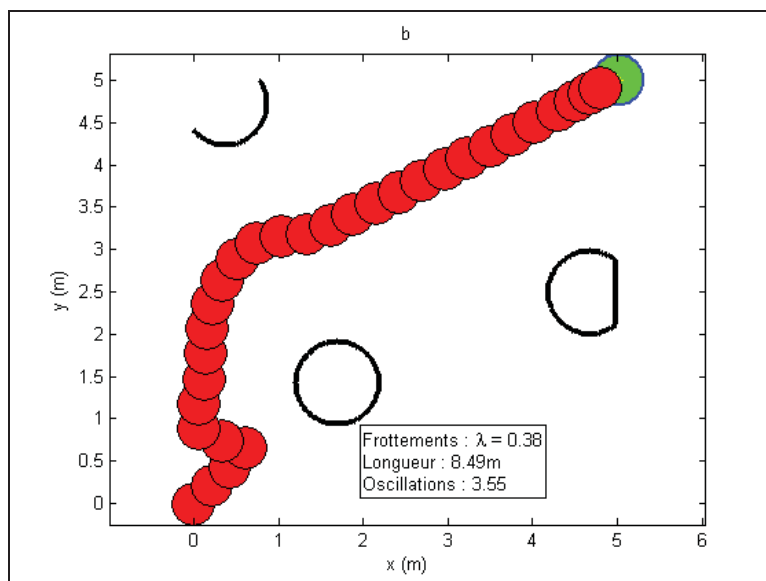
De plus, un effet intéressant est mis en évidence dans la Figure 6-8 : lorsque le robot est pris entre deux obstacles, les forces tangentielles venant de chaque obstacle s'ajoutent, ce qui donne une poussée supplémentaire au robot pour surmonter la barrière virtuelle. Dans ce cas le robot réussit même à retrouver le bon chemin pour atteindre le but, chose qui n'a pas été observée dans les cas précédents. Évidemment, il peut toujours exister des cas où cette poussée n'est pas suffisante pour sortir du minimum local, mais cet avantage n'est pas négligeable.

#### 6.3.1.4. Synthèse

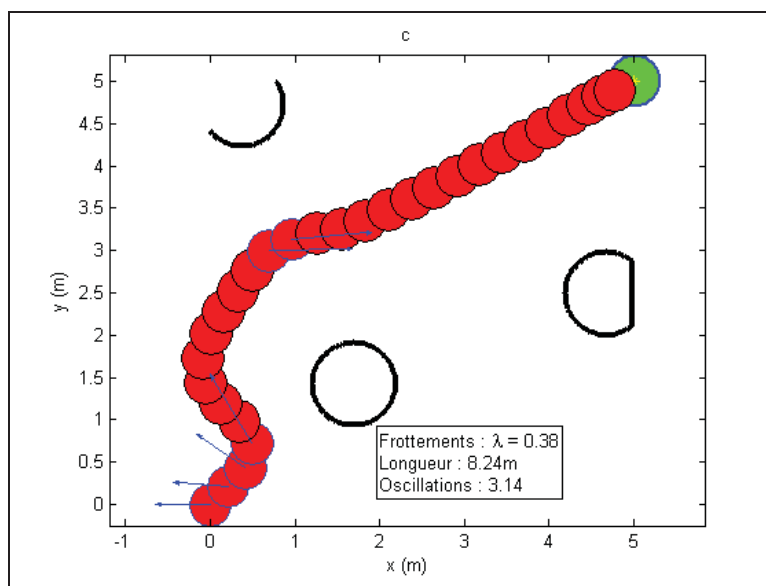
Ici on compare les trois approches précédentes, chacune dans le meilleur des cas.



**Figure 6-11** Forme FIRAS dans le meilleur des cas



**Figure 6-12** Forme améliorée du potentiel de répulsion dans le meilleur des cas



**Figure 6-13** Forme améliorée du potentiel de répulsion avec force de contournement dans le meilleur des cas

Finalement, en comparant les trois cas entre eux, chacun dans ses meilleures conditions, on peut conclure que la nouvelle forme du potentiel de répulsion permet d'améliorer la performance de la navigation à plusieurs niveaux : les oscillations et la longueur de la

trajectoire sont minimales. La durée du parcours a été observée proportionnelle à la distance, ce qui donne une performance comparable du point de vue vitesse.

Tableau 6-1 Comparaison des performances des différentes formes du potentiel de répulsion

<b>Paramètre</b>	<b>FIRAS</b>	<b>Nouvelle forme du potentiel</b>	<b>Nouveau potentiel + Force de contournement</b>
<i>Longueur de la trajectoire <math>L_t</math></i>	9.46	8.49	8.24
<i>Coefficient d'oscillations <math>C_o</math></i>	6.07	3.55	3.14

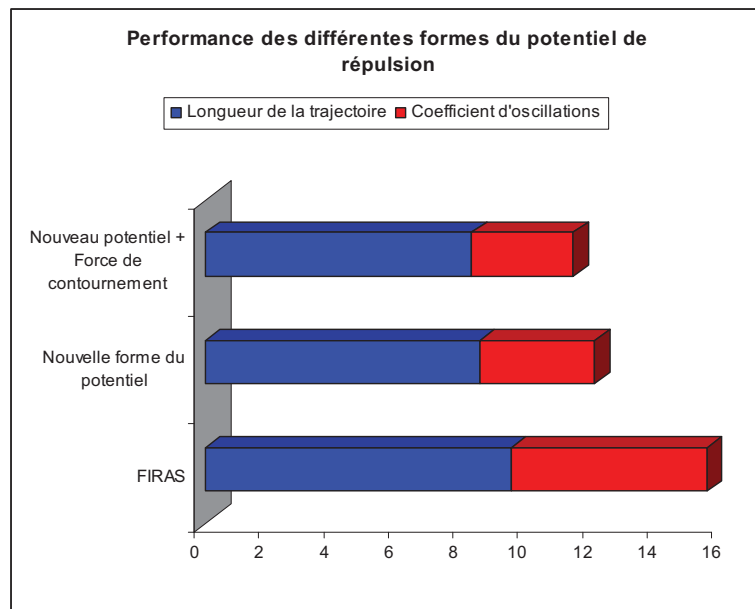


Figure 6-14 Comparaison des performances des différentes formes du potentiel de répulsion

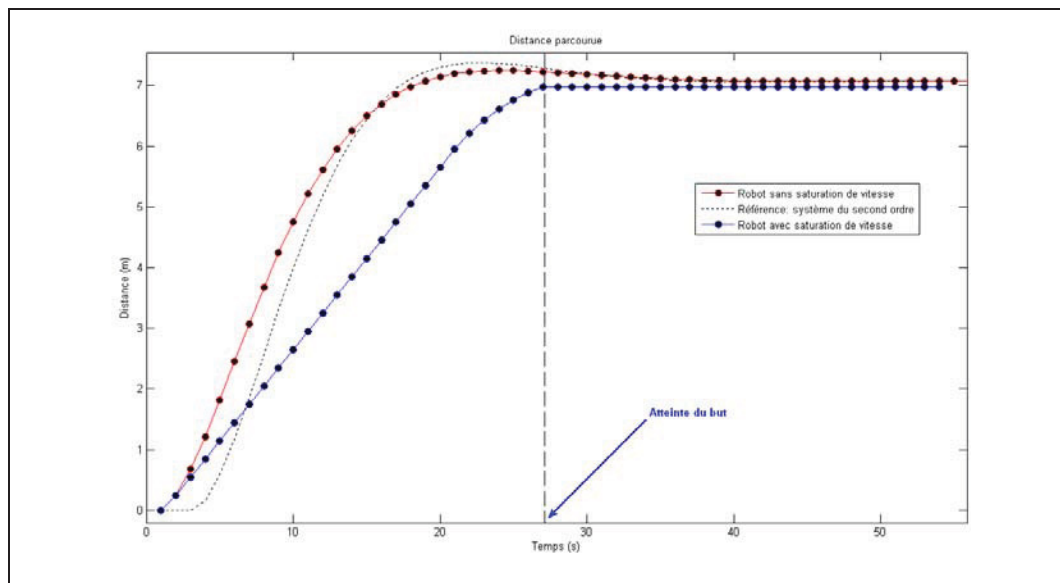
Le Tableau 6-1 et la Figure 6-14 montrent bien que l'utilisation de la nouvelle forme de la force de répulsion entraîne une amélioration notable de la longueur du parcours et du coefficient d'oscillations. De plus, l'utilisation de la force de contournement contribue à réduire encore plus le facteur d'oscillations.

### 6.3.2. Choix objectif des paramètres

Dans ce paragraphe, on montre que le choix des paramètres en utilisant la méthode développée au CHAPITRE 5 donne bien une amélioration des performances. Dans tous les essais avec obstacles, on utilise la nouvelle fonction de répulsion avec la force de contournement.

Sachant que le calcul des paramètres  $\lambda$  et  $\zeta$  par la méthode développée au CHAPITRE 5 suppose qu'il n'y a pas d'obstacles dans l'environnement du robot, on commence par vérifier la validité des critères de calcul de ces paramètres sans obstacles.

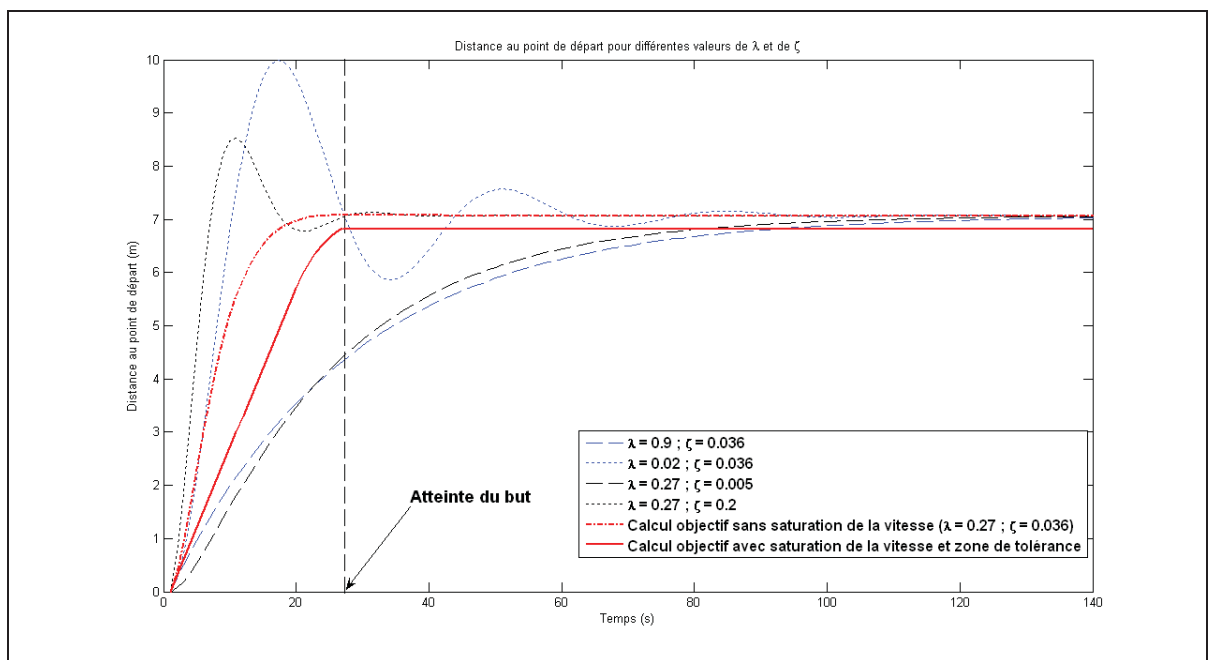
On vérifie d'abord que les paramètres calculés donnent bien au robot un comportement du second ordre. La Figure 6-15 compare le comportement du robot à celui d'un système du second ordre avec les mêmes paramètres de durée et de vitesse moyenne.



**Figure 6-15 Comparaison du comportement du robot avec celui d'un système du second ordre**

On voit bien que le choix des paramètres donne la même performance en temps et vitesse moyenne que le cas idéal.

En variant les paramètres  $\lambda$  et  $\zeta$  autour de leur valeur optimale, on voit une dégradation des performances : Une valeur trop faible de  $\lambda$  ou trop grande de  $\zeta$  rend le robot plus oscillant. Une valeur trop grande de  $\lambda$  ou trop faible de  $\zeta$  rend le robot trop lent (Figure 6-16).



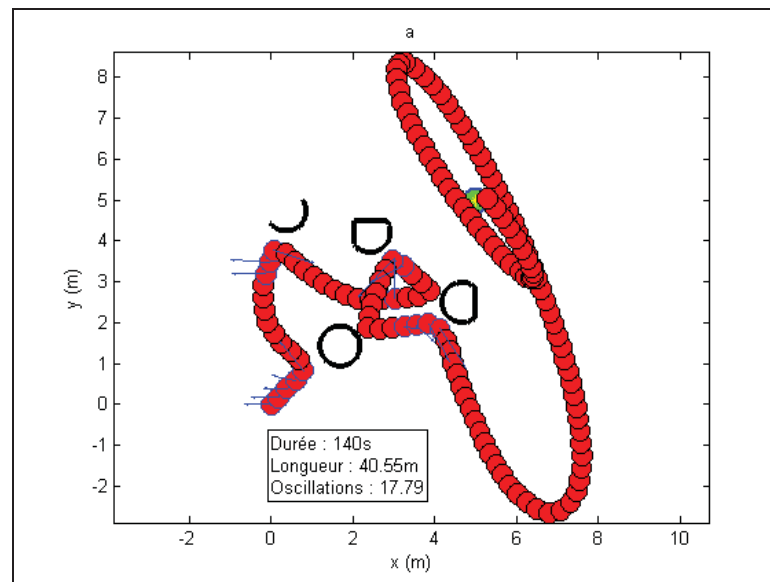
**Figure 6-16 Distance par rapport à l'origine pour différentes valeurs des paramètres  $\lambda$  et  $\zeta$**

En testant l'influence de chaque paramètre séparément, dans un environnement avec obstacles, on peut faire les observations suivantes :

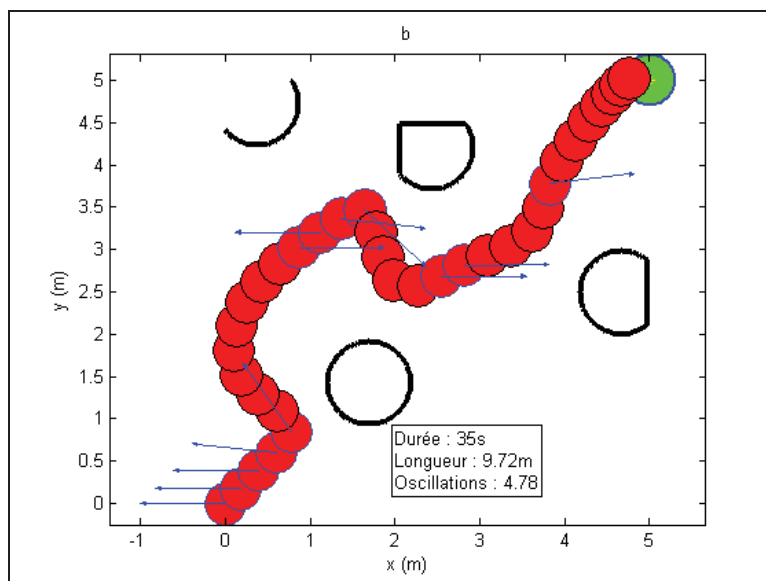
- une valeur faible du paramètre  $\lambda$  tend à rendre la trajectoire plus ample et anguleuse; l'évitement des obstacles est plus brusque. Les valeurs élevées de  $\lambda$  tendent à arrondir les courbes et à rendre la trajectoire plus compacte;
- l'influence du paramètre  $\zeta$  est opposée à celle de  $\lambda$  : les valeurs faibles de  $\zeta$  ont le même effet que les valeurs élevées de  $\lambda$  et vice versa;

- le paramètre  $\eta$  influe sur la capacité du robot à s'approcher des obstacles : de petites valeurs du paramètre empêchent le robot de se rapprocher des obstacles, alors que les grandes valeurs du paramètre permettent au robot de s'approcher des obstacles. Cela a pour effet aussi de réduire le risque d'être piégé dans un minimum local.

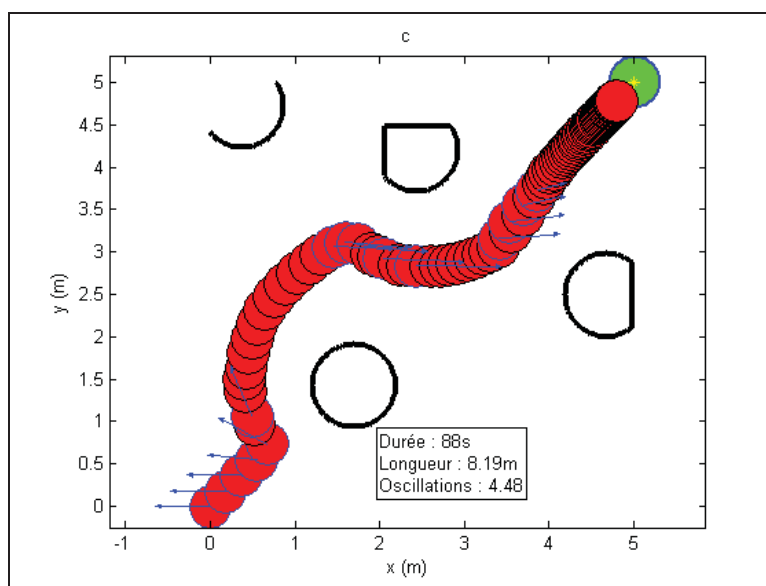
Toutes choses égales par ailleurs, on fait varier chaque paramètre autour de la valeur optimale calculée :



**Figure 6-17** Variation du paramètre  $\lambda$  autour de sa valeur optimale calculée. Valeur trop faible -  $\lambda = 0.05$

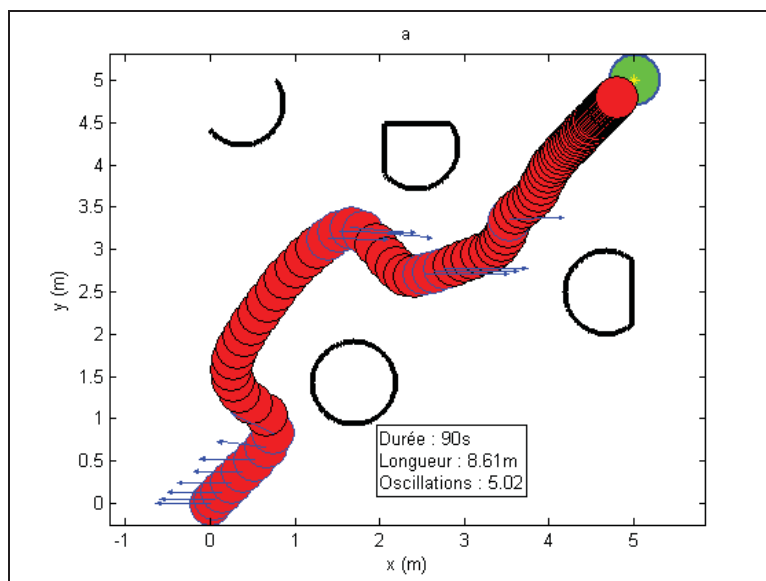


**Figure 6-18** Variation du paramètre  $\lambda$  autour de sa valeur optimale calculée. Valeur calculée -  $\lambda = 0.27$

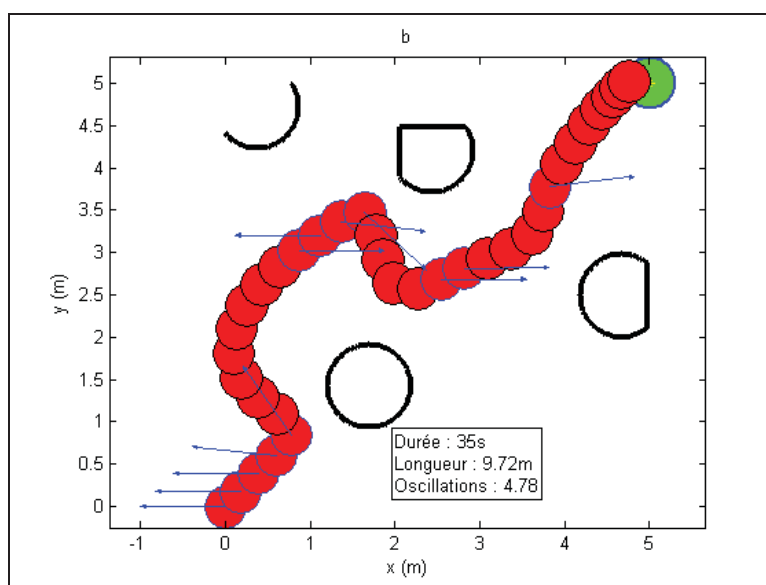


**Figure 6-19** Variation du paramètre  $\lambda$  autour de sa valeur optimale calculée. Valeur trop grande -  $\lambda = 0.9$

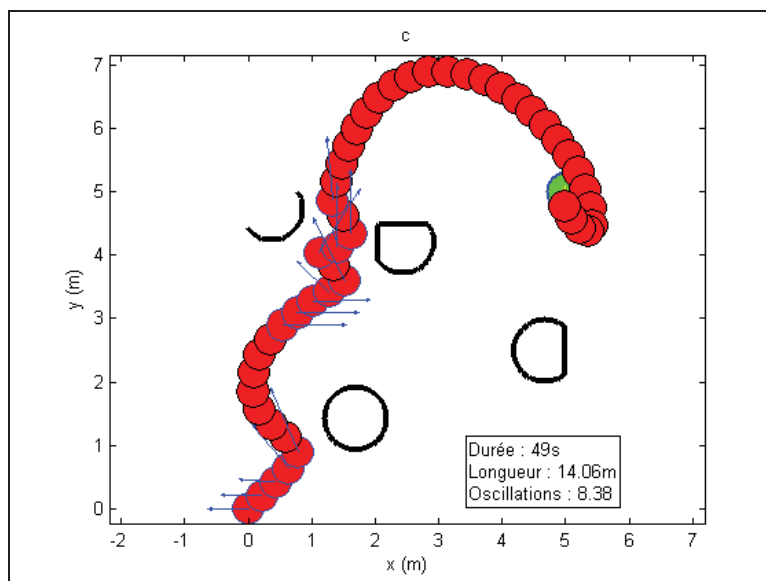




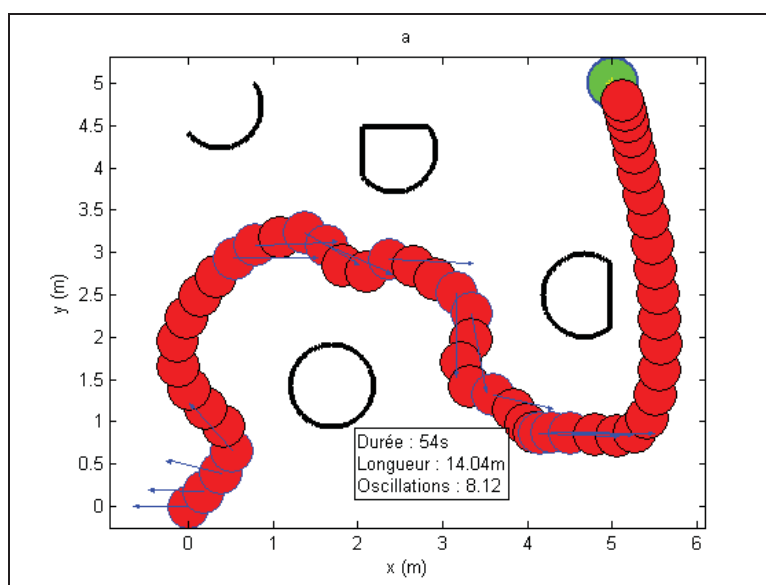
**Figure 6-20** Variation du paramètre  $\zeta$  autour de sa valeur optimale calculée. Valeur trop faible -  $\zeta = 0.01$



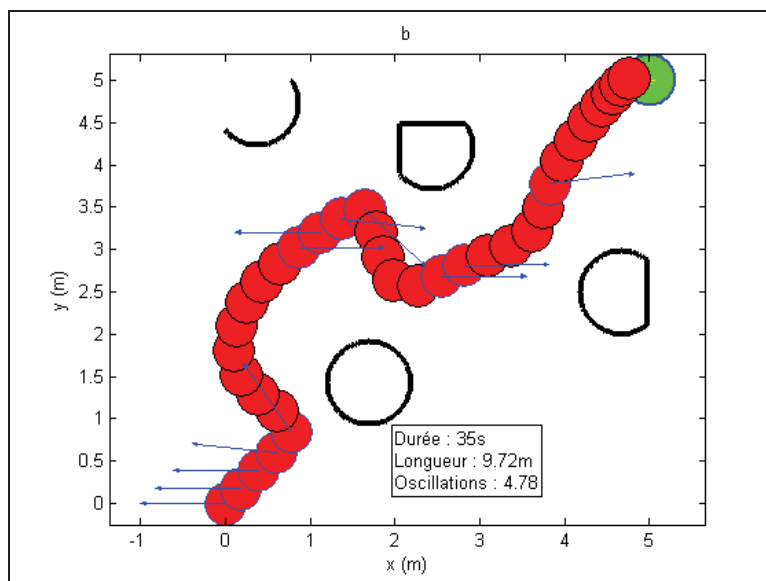
**Figure 6-21** Variation du paramètre  $\zeta$  autour de sa valeur optimale calculée. Valeur calculée -  $\zeta = 0.036$



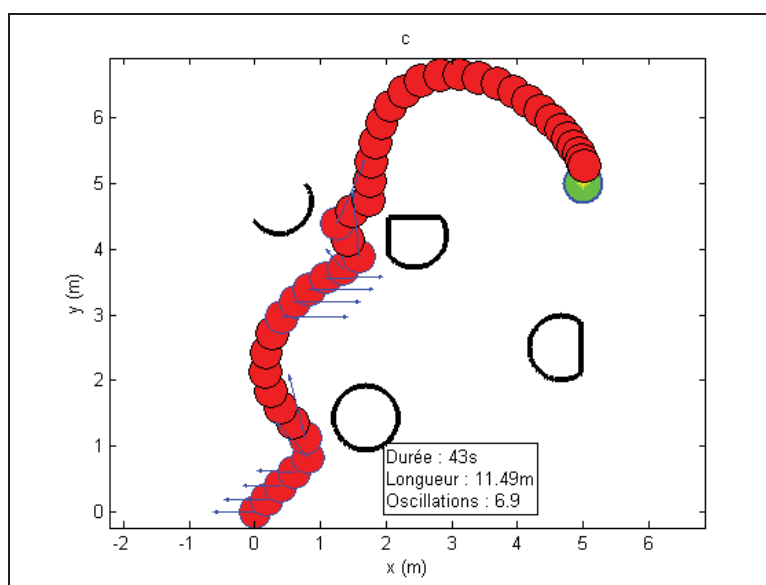
**Figure 6-22** Variation du paramètre  $\zeta$  autour de sa valeur optimale calculée. Valeur trop grande -  $\zeta = 0.2$



**Figure 6-23** Variation du paramètre  $\eta$  autour de sa valeur optimale calculée. Valeur trop faible -  $\eta = 4$



**Figure 6-24** Variation du paramètre  $\eta$  autour de sa valeur optimale calculée. Valeur calculée -  $\eta = 8.44$



**Figure 6-25** Variation du paramètre  $\eta$  autour de sa valeur optimale calculée. Valeur trop élevée -  $\eta = 12$

Afin de comparer les performances de l'algorithme pour différentes valeurs des paramètres, on regarde :

- la longueur de la trajectoire  $L_t$  ;
- la durée du parcours  $D_p$  ;
- le coefficient d'oscillations  $C_o$  .

Pour chacun des essais effectués précédemment, on normalise les performances par rapport à la valeur obtenue lorsque les trois paramètres sont calculés objectivement.

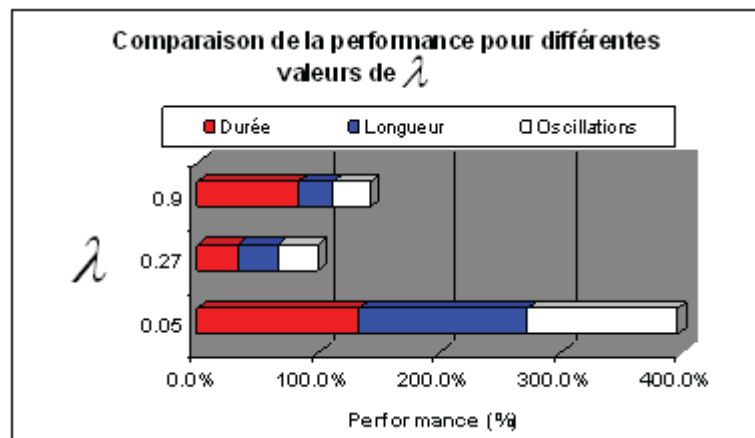


Figure 6-26 Valeur optimale de  $\lambda$

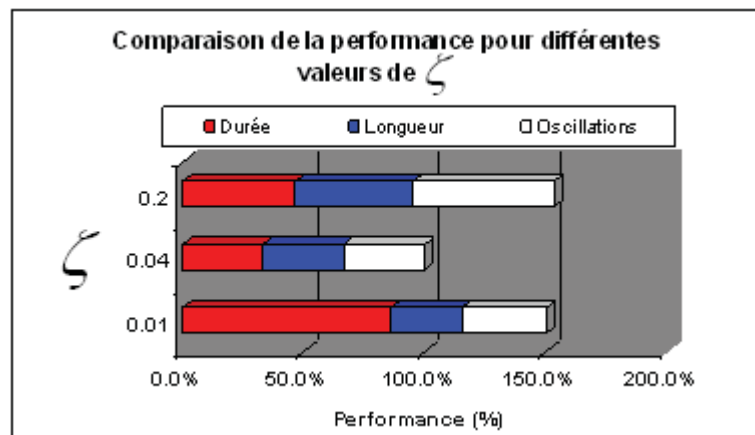
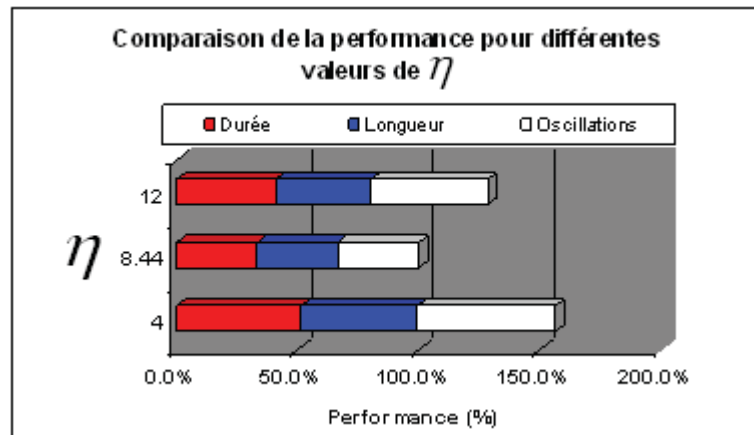


Figure 6-27 Valeur optimale de  $\zeta$



**Figure 6-28 Valeur optimale de  $\eta$**

On voit bien que dans les trois comparaisons, la valeur calculée du paramètre optimise la performance de l'algorithme pour chacun des critères de comparaison.

## 6.4. Résultats pratiques

Dans cette partie on montre la performance de l'algorithme à travers des essais réalisés sur le robot réel.

### 6.4.1. Conditions préliminaires

#### 6.4.1.1. Pas de calcul de l'algorithme

La première contrainte à imposer est le pas de calcul de l'algorithme. En effet, comme vu au paragraphe 2.5, le robot contient une boucle de commande interne qui contrôle la vitesse et le couple des moteurs. Le pas d'échantillonnage de cette boucle de commande interne est de 100ms. Afin de donner le temps au robot de prendre la vitesse que l'algorithme lui impose, il faut s'assurer que la référence de vitesse soit rafraîchie plus lentement que le pas d'échantillonnage de la boucle interne. Nous avons choisi un facteur de 10; le pas de calcul de l'algorithme est donc de 1s.

#### 6.4.1.2. Filtrage du signal des capteurs

Pratiquement, l'algorithme reçoit directement le signal des sonars pour calculer la position des obstacles. Vu que ce genre d'instrument est extrêmement bruité, il est nécessaire de filtrer le signal. Ce signal est généré chaque 100ms par la boucle de commande interne du robot. Vu que l'algorithme a un pas de calcul de 1s, il est possible d'effectuer une moyenne des 10 signaux pour chaque sonar. On réduit ainsi le bruit du signal des capteurs. Il faut noter que même après filtrage, le signal des sonars est très bruité; cela sera illustré par les résultats.

#### 6.4.1.3. Discrétisation de l'équation du mouvement

Les équations utilisées dans le calcul des différentes grandeurs dans l'algorithme sont toutes algébriques, à l'exception de celle donnant la référence de vitesse. En effet, l'équation de la dynamique fait intervenir l'accélération et la vitesse (à cause du facteur de frottements virtuels). Afin de résoudre cette équation numériquement, il est nécessaire de la discrétiser. La solution la plus simple consiste à approximer la dérivée par une différence de degré 1 :

$$a = \frac{dV}{dt} \equiv \frac{V_k - V_{k-1}}{\tau} \quad (6.6)$$

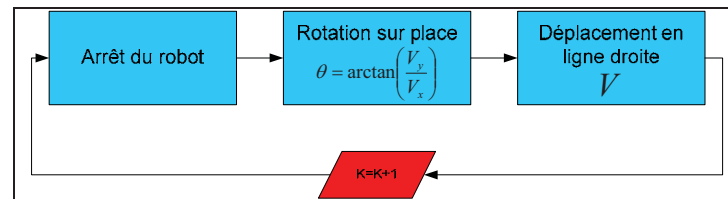
Où  $\tau$  représente le pas de calcul de l'algorithme, c'est-à-dire ici 1s.

Cette approximation simple permet de minimiser la puissance de calcul nécessaire. Les résultats montreront que cette discrétisation est suffisante.

#### 6.4.1.4. Adaptation de l'algorithme à la cinématique du robot

L'algorithme développé jusqu'à présent considère que le robot est un point matériel. Or pratiquement, la cinématique du robot n'est pas libre et la contrainte non holonome impose des contraintes à ajouter. Il est donc nécessaire de concevoir une boucle de commande du robot pour qu'il puisse bien suivre la référence. Plusieurs approches sont possibles –

Nasrallah (2007). Dans notre cas, on cherche à minimiser la complexité du calcul en utilisant une particularité de la cinématique du robot utilisé. On applique une loi de commande qui consiste à faire suivre au robot une trajectoire par segments; à chaque pas de calcul de l'algorithme, le robot s'arrête. Une fois le vecteur vitesse de référence est calculé, le robot tourne sur place pour prendre la bonne direction. Lorsqu'il atteint cette direction, le robot se déplace alors en ligne droite jusqu'à atteindre le point projeté. En résumé, la méthode de commande du robot serait illustrée dans la Figure 6-29.



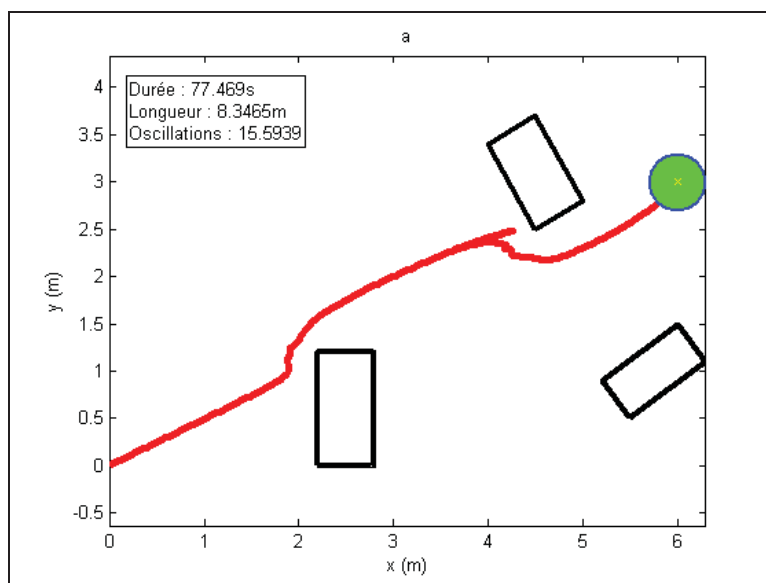
**Figure 6-29 Adaptation de l'algorithme à la cinématique du robot**

#### 6.4.2. Efficacité de la force de contournement

Dans cette partie on montre que l'ajout de la force de contournement réussit à améliorer la trajectoire du robot selon les critères  $C_o$ ,  $L_t$  et  $D_p$  fixés précédemment.

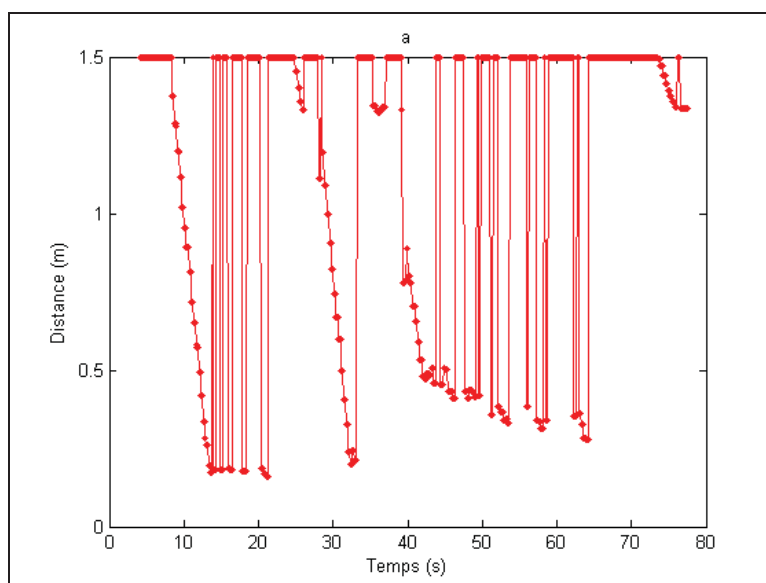
En choisissant les mêmes valeurs des paramètres et en faisant l'expérience sans utiliser la force de contournement, puis en utilisant la force de contournement, on voit que la trajectoire réelle semble plus intéressante lorsque la force de contournement est utilisée.

En effet, sans utiliser la force de contournement, la Figure 6-30 montre que le robot a tendance à osciller. Il est intéressant de noter aussi que lorsque le robot approche le deuxième obstacle, sa direction est perpendiculaire à la surface de cet obstacle. On voit bien que le robot rebrousse chemin et effectue plusieurs oscillations pour contourner cet obstacle. Ceci illustre le problème d'oscillations lorsque le robot se dirige perpendiculairement à la surface de l'obstacle. C'est dans ces conditions précisément que la force de contournement devrait jouer un rôle important.



**Figure 6-30** Forme améliorée du potentiel de répulsion - sans force de contournement

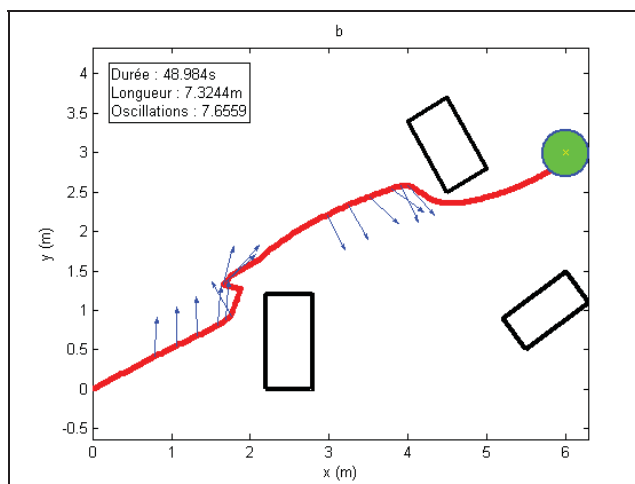
La Figure 6-31 illustre les oscillations lorsque le robot s'approche d'un obstacle. Les oscillations élevées de la distance par rapport à l'obstacle sont évidentes lorsque le robot approche le second obstacle.



**Figure 6-31** Distance de l'obstacle le plus proche - sans force de contournement

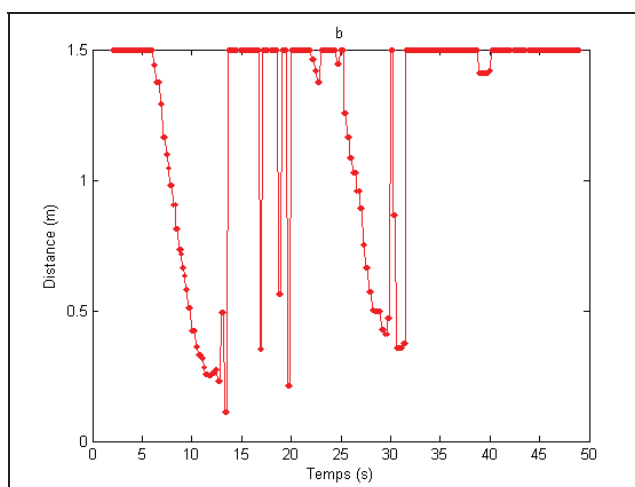


Dans la Figure 6-32 on voit bien comment, lorsque le robot approche le second obstacle, la force de contournement entre en jeu pour éliminer les oscillations.



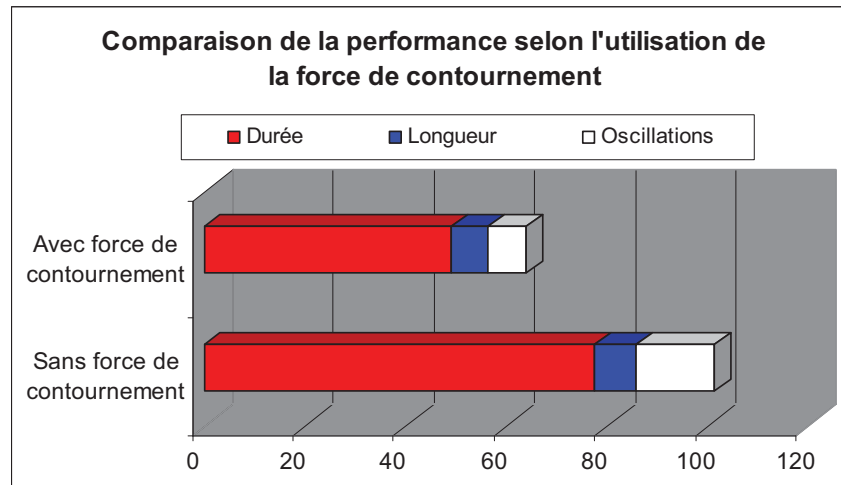
**Figure 6-32** Forme améliorée du potentiel de répulsion - avec force de contournement

Lorsque le robot est proche du premier obstacle, une certaine irrégularité de la trajectoire est observée. Cette irrégularité est explicable par le fait que le robot se trouve en point anguleux de la surface de l'obstacle. Ceci accentue la vulnérabilité des sonars au bruit. La Figure 6-33 montre bien cette indétermination aux alentours de l'instant 13s.



**Figure 6-33** Distance de l'obstacle le plus proche - avec force de contournement

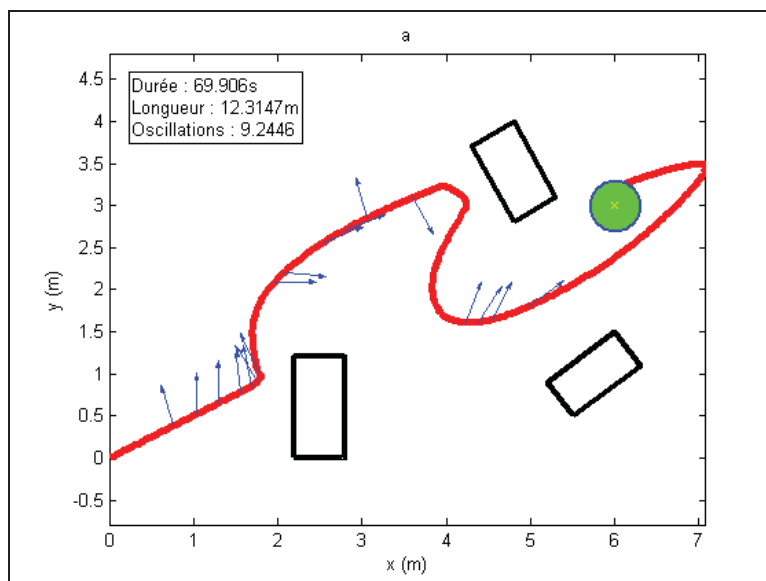
En comparant les deux trajectoires on voit que la durée, la longueur ainsi que le coefficient d'oscillations sont tous les trois plus faibles lorsque la force de contournement est utilisée.



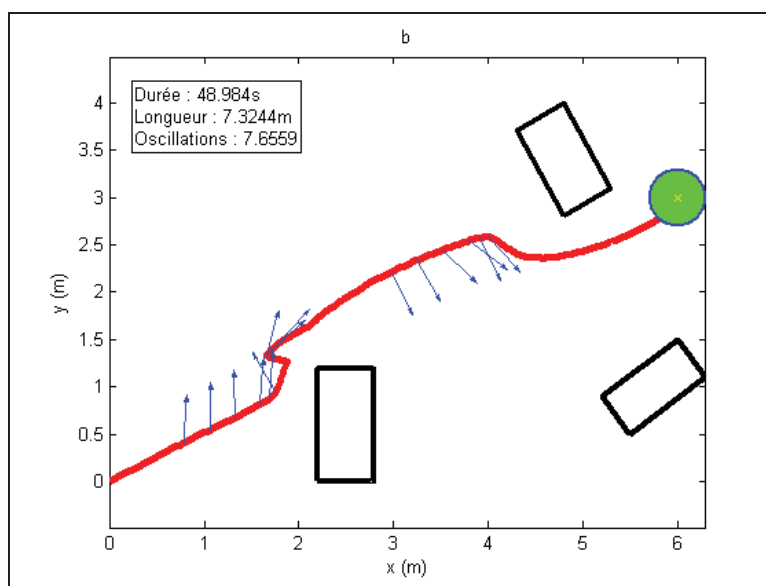
**Figure 6-34 Comparaison des performances selon l'utilisation de la force de contournement**

#### 6.4.3. Choix objectif des paramètres

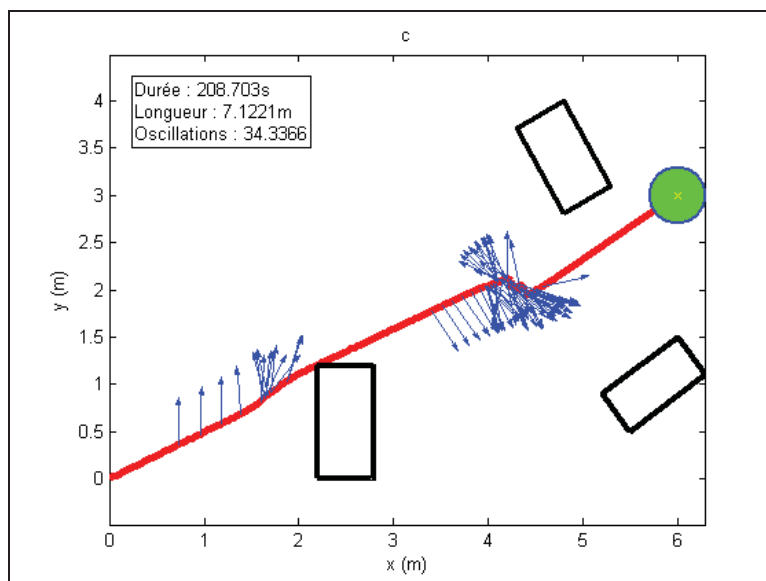
Ici on illustre l'influence des paramètres  $\lambda$ ,  $\zeta$  et  $\eta$ . On effectue des essais en variant à chaque fois un paramètre en gardant les autres inchangés.



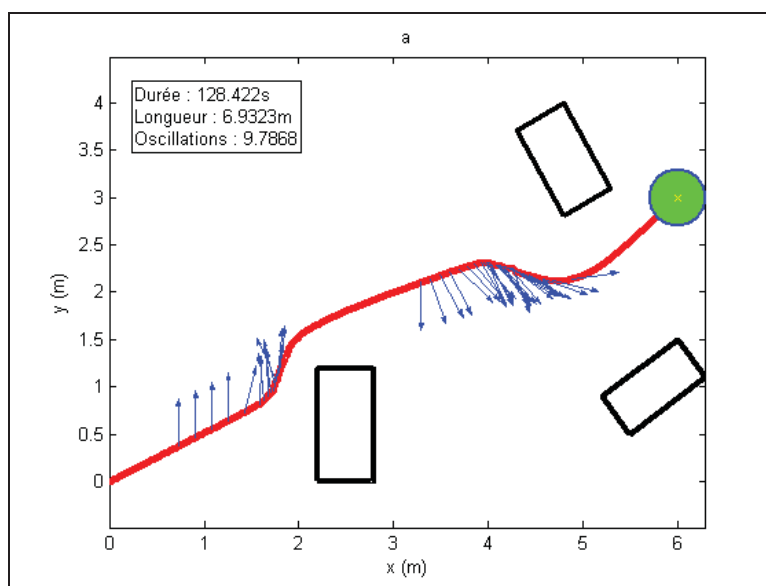
**Figure 6-35** Variation du paramètre  $\lambda$  autour de sa valeur optimale calculée. Valeur trop faible -  $\lambda = 0.1$



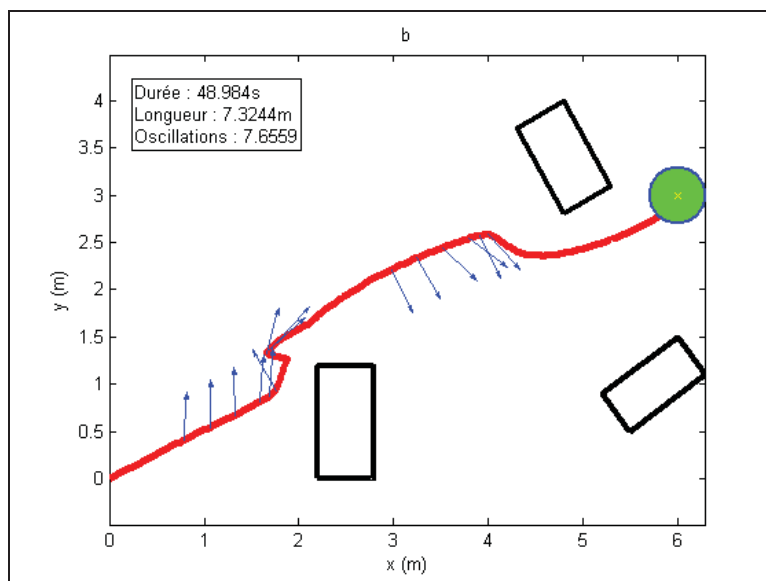
**Figure 6-36** Variation du paramètre  $\lambda$  autour de sa valeur optimale calculée. Valeur calculée -  $\lambda = 0.27$



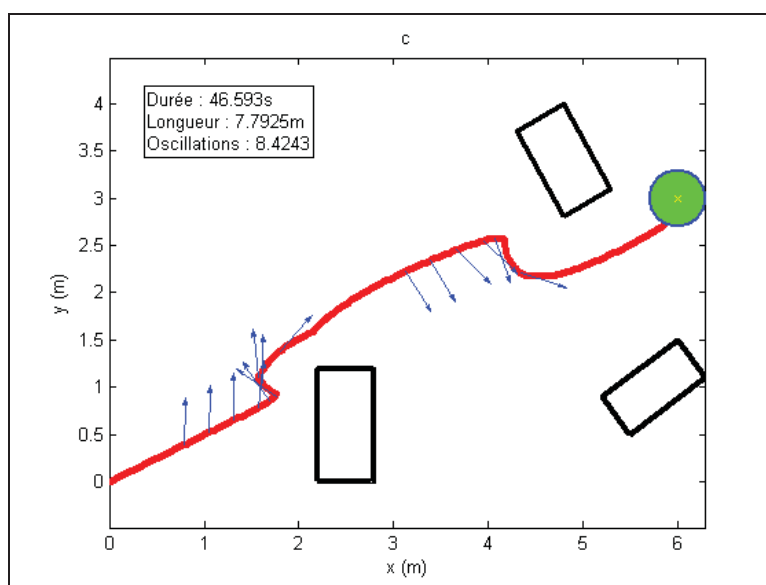
**Figure 6-37** Variation du paramètre  $\lambda$  autour de sa valeur optimale calculée. Valeur trop élevée -  $\lambda = 0.9$



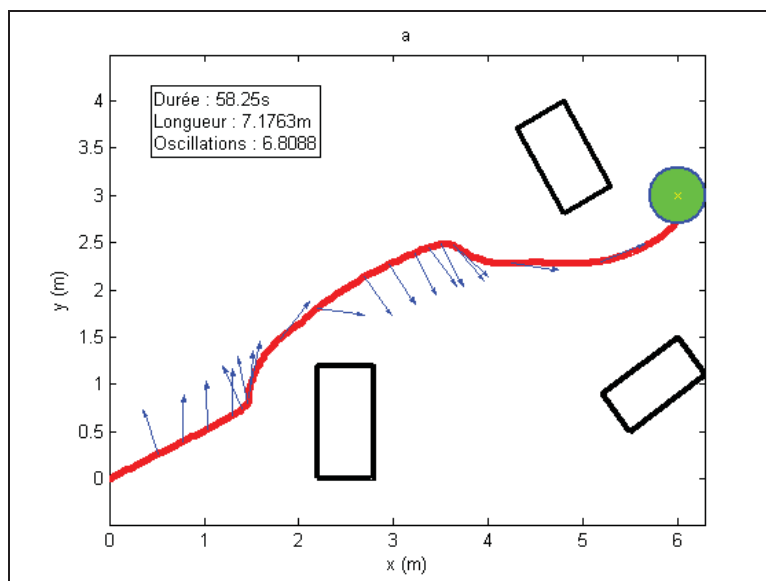
**Figure 6-38** Variation du paramètre  $\zeta$  autour de sa valeur optimale calculée. Valeur trop faible -  $\zeta = 0.01$



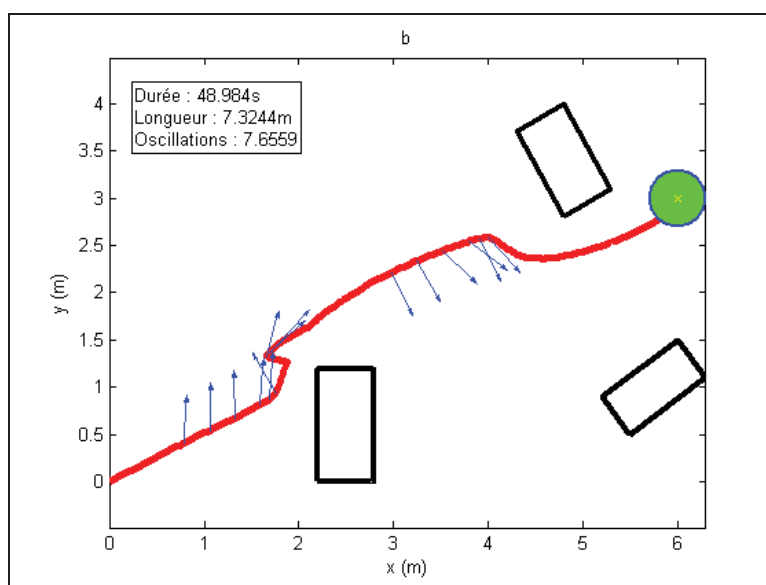
**Figure 6-39** Variation du paramètre  $\zeta$  autour de sa valeur optimale calculée. Valeur calculée -  $\zeta = 0.036$



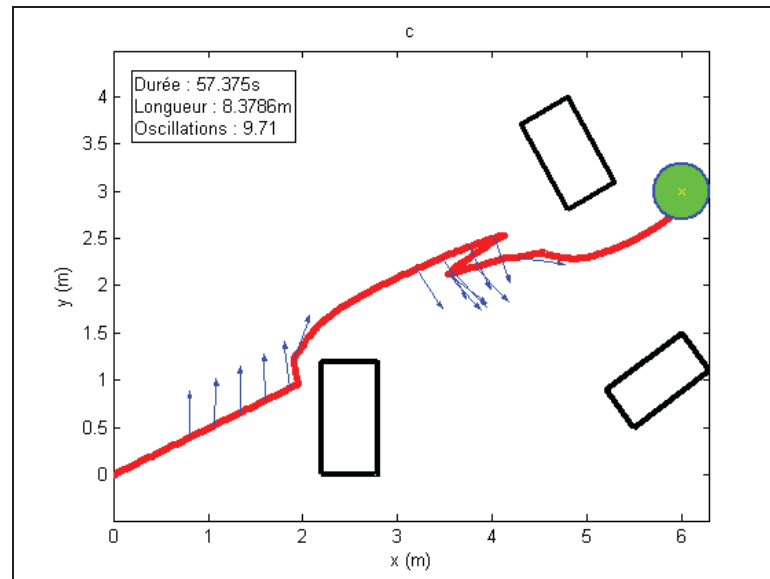
**Figure 6-40** Variation du paramètre  $\zeta$  autour de sa valeur optimale calculée. Valeur trop élevée -  $\zeta = 0.2$



**Figure 6-41** Variation du paramètre  $\eta$  autour de sa valeur optimale calculée. Valeur trop faible -  $\eta = 4$

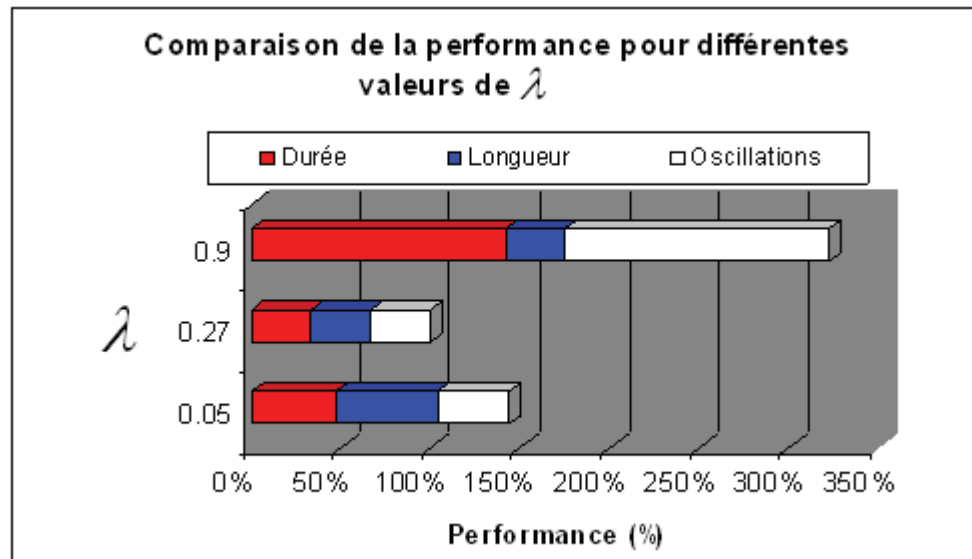
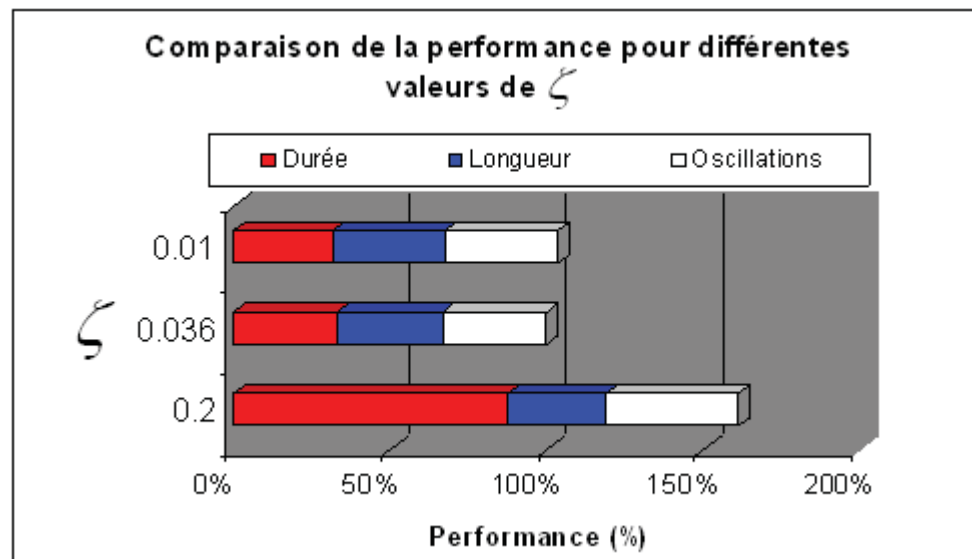


**Figure 6-42** Variation du paramètre  $\eta$  autour de sa valeur optimale calculée. Valeur calculée -  $\eta = 8.44$

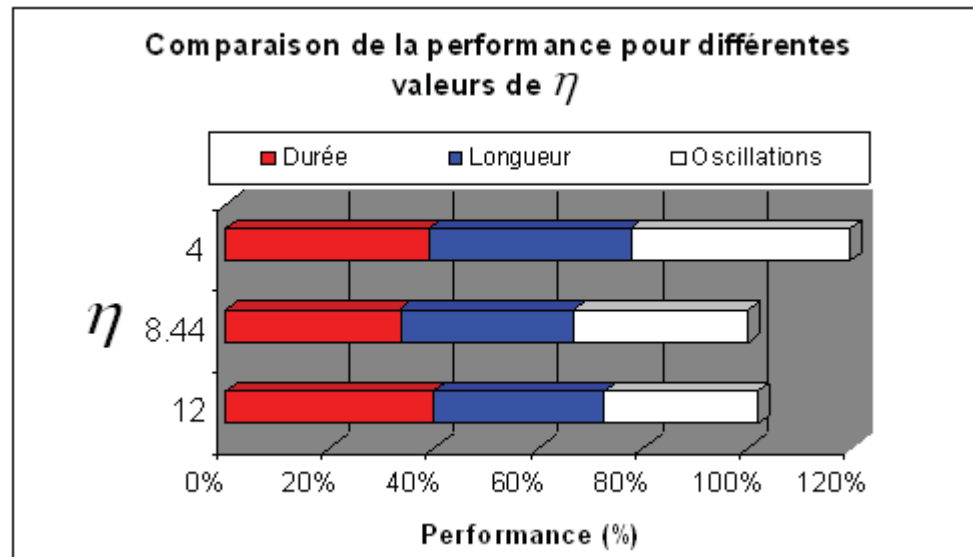


**Figure 6-43** Variation du paramètre  $\eta$  autour de sa valeur optimale calculée. Valeur trop élevée -  $\eta = 12$

La comparaison des valeurs normalisées des paramètres de performance montre que, tout comme dans le cas des simulations, le calcul objectif des paramètres des potentiels tend bien à optimiser la performance. Notons qu'à la Figure 6-37, le robot passe très proche de l'obstacle. L'épaisseur des lignes sur le graphe donne l'illusion d'un contact.

Figure 6-44 Valeur optimale de  $\lambda$ Figure 6-45 Valeur optimale de  $\zeta$





**Figure 6-46 Valeur optimale de  $\eta$**

Notons que le bruit des sonars tend à estomper l'avantage des paramètres calculés, sans toutefois l'éliminer.

## 6.5. Conclusion

Finalement des essais ont été faits dans ce chapitre pour tester les divers aspects traités en théorie.

Certaines contraintes relatives à l'utilisation d'un robot réel ont été imposées afin de pouvoir adapter la théorie qui s'applique à un point matériel au robot qui a une cinématique liée.

L'avantage de l'utilisation de la force de contournement a été mis en évidence à la fois par les résultats de simulation et les résultats pratiques. La validité du calcul objectif a aussi été appuyée par les résultats des simulations et du robot réel.

## CONCLUSION

### Synthèse

Nous avons synthétisé dans ce projet une méthode complète de navigation pour un robot mobile. La méthode développée a permis d'améliorer les performances de la navigation dans les mêmes conditions d'environnement. Une méthode d'initialisation automatique des paramètres a aussi été introduite. Le modèle et l'architecture du robot utilisé ont été détaillés.

La méthode développée s'est basée sur des principes et approches connues pour en ressortir une structure générale dont les performances dépassent celles des méthodes de base.

En effet, les limitations de la méthode classique des champs de potentiels artificiels, en l'occurrence le problème d'oscillation, celui du *GNRON*, ainsi que celui des minima locaux, ont été identifiés. À l'exception du problème relié aux minima locaux qui n'a pas été abordé dans ce projet, faute de temps, les limitations reliées à la méthode classique des champs de potentiels ont été minimisées, voire éliminées.

De plus, les méthodes de base ont tendance à fixer les paramètres par essai-erreur au lieu d'utiliser une méthode de calcul précise pour trouver les valeurs adéquates des paramètres. Nous avons donc développé une méthode de calcul objectif des paramètres en se fixant des critères d'optimisation.

Les principales affirmations et équations ont été extensivement testées et comparées à celles obtenues avec la méthode classique.

### Contribution

Ce projet a contribué à l'amélioration de la navigation utilisant les champs de potentiels sur les plans suivants.

### ***Minimisation des oscillations de la trajectoire***

La force de répulsion des obstacles est, dans le cas classique, trop simpliste du point de vue flexibilité. En effet, la seule action possible du robot au voisinage d'un obstacle est de s'en éloigner le plus vite possible (force radiale centrifuge). L'ajout d'une force de contournement des obstacles a permis d'ajouter un degré de liberté supplémentaire et de lisser la trajectoire autour des obstacles en considérant un évitement moins simpliste qui stabilise la trajectoire en modulant l'action du robot entre évitement et contournement.

### ***Développement d'une méthode objective de choix des paramètres***

Le choix des paramètres et de la forme des équations du potentiel ne prenait pas en considération les oscillations de la trajectoire et il n'y avait pas de méthode objective de choix optimal des paramètres. Une méthode objective basée sur des critères d'optimisation a été introduite, de façon à améliorer les oscillations sans passer par une période de réglage des paramètres par essai-erreur.

### ***Navigation en environnement inconnu en utilisant peu de ressources***

La méthode développée a été appliquée au robot P3AT équipé d'une multitude de capteurs et de moyens importants. Cependant, la méthode a été pensée de façon à utiliser le moins de ressources possibles pour être applicable à tout robot muni d'un minimum de capteurs.

De plus, cette méthode ne prend en compte aucune connaissance préalable de l'environnement; la trajectoire est générée au fur et à mesure que le robot avance et découvre son environnement immédiat. Bien entendu, il existe des méthodes de navigation sans connaissance préalable de l'environnement, mais utilisant des moyens de calcul importants ou des capteurs encombrants ou coûteux. La méthode développée s'est limitée aux moyens de

bord primitifs sans toutefois réduire la capacité à naviguer en environnement inconnu, ou même dynamique.

### *Expérimentation extensive*

La méthode développée a été mise à l'épreuve à travers des essais sous plusieurs conditions afin de la comparer à la méthode classique de champs de potentiels. La performance de la méthode a été prouvée par des résultats de simulations ainsi que par des tests sur un robot réel, dans des conditions non idéales d'expérimentation.

## RECOMMANDATIONS

Bien que la méthode introduite dans ce projet ait donnée des résultats satisfaisants, certains points intéressants restent à explorer.

### **Intégration de la méthode de résolution des minima locaux**

La méthode introduite a permis de réduire l'impact des limitations des champs de potentiels. En effet, les problèmes d'oscillations et du GNRON ont été résolus. Cependant, aucune méthode de sortie des minima locaux n'a été intégrée. Il serait fort intéressant d'intégrer une méthode de sortie des minima locaux, ce qui rendra la méthode complète en garantissant que le robot atteindra toujours le but, même en environnement inconnu.

### ***Développement d'une interface homme/machine***

La navigation du robot P3AT a été faite en utilisant une application du type console permettant d'envoyer des commandes directement au robot, en permettant une interaction limitée de l'utilisateur. Il serait très intéressant de développer une interface complète qui permet d'intégrer toutes les fonctions du robot afin d'en tirer tous les avantages.

Une application client pourrait être développée pour permettre à l'utilisateur de communiquer avec le robot par exemple en visualisant le signal de la caméra embarquée, en envoyant les coordonnées de façon plus conviviale et plus dynamique, en visualisant la position et le signal des capteurs en temps réel... Une application serveur devrait aussi être développée pour prendre en charge les commandes de haut niveau venant du client.

## BIBLIOGRAPHIE

- Araujo, A. F. R., D'Arbo, H. Jr. 1998. « A Partially Recurrent Neural Network to Perform Trajectory Planning, Inverse Kinematics, and Inverse Dynamics ». *IEEE International Conference on Systems, Man, and Cybernetics*, Volume 2, Oct. 1998, pp. 1784-1789.
- Bertsekas, Dimitri P. 1999. *Nonlinear Programming*. 2<sup>nd</sup> Edition. Athena Scientific, 802p.
- Chang, T. Y., Kuo, S. W., Hsu, J. Y. J. 1994. « A Two-Phase Navigation System for Mobile Robots in Dynamic Environments ». *Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems*, vol. 1, Sept. 1994, pp. 306-313
- Ge, S. S., Cui, Y. J. 2000. « New Potential Functions for Mobile Robot Path Planning » *IEEE Transactions on Robotics and Automation*, vol. 16, no.5, Oct. 2000, pp. 615-620
- Jianping, T., Yang, S.X. « Genetic Algorithm Based Path Planning for a Mobile Robot ». In *Proceedings of the IEEE International Conference on Robotics and Automation* (Taipei, Taiwan, 14-19 Sept. 2003) pp. 1221-1226 vol.1.
- Khatib, O. 1986. « Real-time obstacle avoidance for manipulators and mobile robots » *International Journal of Robotic Research*, vol. 5, n°1, pp. 90-98
- Khosla, P., Volpe, R. 1998. « Superquadric artificial potentials for obstacle avoidance and approach », *Proceedings of the IEEE International Conference on Robotics and Automation*, (Philadelphia, PA, April 26-28)
- Koh, K. C., Cho, H. S. 1995. « Wheel Servo Control Based on Feedforward Compensation for an Autonomous Mobile Robot ». *Proceedings of the International Conference on Intelligent Robots and Systems 95 : Human Robot Interaction and Cooperative Robots*.
- Krogh, B. 1984. « A Generalized Potential Field Approach to Obstacle Avoidance Control ». In *Proceedings of the ASME Conference on Robotic Research : The Next Five Years and Beyond*. Bethlehem, Pennsylvania, 1984.
- Lee, Leng-Feng. 2004. « Decentralized Motion Planning Within an Artificial Potential Framework (APF) for Cooperative Payload Transport by Multi-Robot Collectives ». *Mémoire de Maîtrise*, Buffalo, New York, State University of New York at Buffalo, 193p.
- Maalouf, Élie. 2005. « Nonlinear Control of Wheeled Mobile Robots ». *Mémoire de maîtrise*, Montréal, École de Technologie Supérieure. 165p.

- Nasrallah, Danielle. 2007. « Contributions to the Modelling and Control of Two-Wheeled Mobile Robots ». Thèse de Doctorat. Montréal, McGill University. 168p.
- Nise, Norman. 2003. *Control Systems Engineering*, 4<sup>th</sup> Edition. Wiley, 983p.
- Park, M. G., Lee, M. C. 2003. « Artificial Potential Field Based Path Planning for Mobile Robots using a Virtual Obstacle Concept ». In *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. (Kobe, Japan, 20-23 Jul. 2003), pp. 735-740
- Prusky, Alain. 1996. *Robotique mobile : la planification de trajectoire*. Collection « Série robotique ». Paris : Hermes, 236p.
- Ren, J., McIsaac, K. A., Patel R. V. 2006. « Modified Newton's Method Applied to Potential Field-Based Navigation for Mobile Robots ». *IEEE Transactions on Robotics*, vol. 22, no.2, Apr. 2006, pp. 384-391
- Sfeir J., Saad M., Saliyah-H. H. « Navigation d'un robot mobile en environnement inconnu en utilisant les champs de potentiels artificiels. » I2LORNET (Montréal, Québec, 5,6 Octobre 2007).
- Siemiatkowska, B. 1994. « Cellular Neural Networks for Mobile Robot Navigation », Third IEEE International Workshop on Cellular Neural Networks and Their Applications, Dec. 18-21
- Stentz, A. 1994. « Optimal and efficient path planning for partially-known environments ». In *Proceedings of the IEEE International Conference on Robotics and Automation* (San Diego, California, 8-13 Mai 1994) pp. 3310-3317 vol.4.
- Xu, H., Yang, S. X. 2001. « Tracking Control of a Mobile Robot with Kinematic and Dynamic Constraints ». *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*.
- Yang, S. X., Meng, M. 2001. « Neural Network Approaches to Dynamic Collision-Free Trajectory Generation ». In *IEEE Transactions on Systems, Man, and Cybernetics, Part B : Cybernetics*, vol. 31, n° 3, pp. 302-318.
- Zegers, P., Sundareshan, M. K. 2003. « Trajectory Generation and Modulation Using Dynamic Neural Networks ». *IEEE Transactions on Neural Networks*, vol. 14, n°4, pp. 520-533